

UNIVERSIDAD
CARLOS III DE MADRID

ESCUELA POLITÉCNICA SUPERIOR
DEPARTAMENTO DE TEORÍA DE LA SEÑAL Y COMUNICACIONES



PROYECTO FIN DE CARRERA

Tutorial Web de Técnicas de Digitalización de
Audio para la asignatura Tratamiento Digital de
Audio

INGENIERÍA TÉCNICA DE TELECOMUNICACIÓN:
SONIDO E IMAGEN

Autor:

FRANCISCO JAVIER MARTÍNEZ MENCÍAS

Tutor:

SERGIO SANZ RODRÍGUEZ-ESCALONA

Leganés, mayo de 2009

Datos generales del Proyecto

Título: TUTORIAL DE TÉCNICAS DE DIGITALIZACIÓN DE AUDIO PARA LA ASIGNATURA TRATAMIENTO DIGITAL DE AUDIO

Autor: FRANCISCO JAVIER MARTÍNEZ MENCÍAS

Tutor: SERGIO SANZ RODRÍGUEZ-ESCALONA

Tribunal

Presidente: JOSÉ JESÚS GARCÍA RUEDA

Secretario: EDUARDO MARTÍNEZ ENRÍQUEZ

Vocal: LUIS AZPICUETA RUIZ

Resumen del Proyecto

El objetivo del presente proyecto fin de carrera es el desarrollo de un Tutorial Web para la asignatura de Tratamiento Digital de Audio, impartida en el tercer curso de Ingeniería Técnica de Telecomunicación, especialidad Sonido e Imagen. Este tutorial aborda las técnicas de digitalización más comunes aplicadas en audio digital para mejorar la relación señal – ruido de la secuencia digitalizada, que son, concretamente: sobremuestreo, dither y Noise-Shaping. Se expondrán, por tanto, los conceptos teóricos de la materia en combinación con aplicaciones interactivas que faciliten la experimentación práctica de las técnicas estudiadas.

Calificación

Realizado el acto de presentación y defensa del Proyecto Fin de Carrera el día 22 de mayo de 2009, en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, se acuerda otorgar la calificación de:

MATRÍCULA DE HONOR

Agradecimientos

Este proyecto y en realidad, la carrera universitaria que culmina con él, ha sido posible que finalice con éxito gracias a las personas que durante todo este tiempo me han rodeado y que merecen ser reconocidas. Quiero expresar mi más sincero agradecimiento a todos ellos, puesto que humildemente, aunque el trabajo personal y el esfuerzo sean necesarios, el apoyo de todos ellos ha sido fundamental.

En primer lugar, mis padres, José Manuel y Lourdes, que son los responsables de que haya llegado hasta aquí, ayudándome y guiándome en todo momento, por el camino correcto. Quiero dedicárselo especialmente a mi hermano pequeño Alejandro, a quien ahora le toca su turno en los estudios, deseándole mucha suerte; y a mis abuelos Isabelo y Benedicta, a quienes tengo muchísimo cariño y me ilusiona darles esta satisfacción.

Mis amigos del instituto, de mi barrio, que han sido siempre importantísimos para mí, han entendido las dificultades por las que he pasado y siempre me han animado a seguir adelante, aún cuando los exámenes y las clases me hayan quitado tanto tiempo.

Mis compañeros de clase, de la universidad, quienes recordaré siempre, me han aportado muchas cosas y guardo de ellos una gran amistad; juntos, tras tantas horas de prácticas, biblioteca, trabajos, hemos conseguido llegar al final y superar esta etapa.

También, quiero mencionar a mis jefes y compañeros de trabajo, que durante el tiempo en el que empecé mi primera experiencia laboral y terminaba este proyecto, fueron una gran fuente de consejos y ánimo.

Finalmente, como no, a Sergio, el tutor de este proyecto, por haber depositado su confianza en mí y aportarme ideas muy valiosas y además, haber tenido la paciencia suficiente para ayudarme a corregir mis errores y salvar las dificultades.

A todos ellos, gracias

Índice de contenidos

1	Introducción	13
1.1	Presentación del proyecto. Motivación y objetivos.....	13
1.2	Tele-educación y e-learning.....	16
1.3	Estado del arte.....	22
2	Conceptos teóricos	33
2.1	Procesado digital de señal	33
2.1.1	Análisis espectral de señal.....	34
2.1.1.1	El análisis armónico de señales.....	36
2.1.1.2	La Transformada de Fourier	37
2.1.1.3	Densidad Espectral de Potencia y caracterización de procesos estocásticos.....	39
2.1.1.4	La Transformada Rápida de Fourier (FFT).....	43
2.1.1.5	Enventanado de señal	46
2.1.1.6	Cálculo de la DEP promedio. El método de Welch	50
2.1.2	Filtrado de señal	52
2.1.2.1	Concepto.....	52
2.1.2.2	Estructura de los filtros digitales.....	55
2.1.2.3	Aplicación de los filtros en las aplicaciones y diseño.....	57
2.2	Técnicas de digitalización de audio.....	62
2.2.1	Conversión analógico / digital (A / D).....	62
2.2.1.1	Desarrollo teórico del proceso de conversión A / D	63
2.2.1.2	Digitalización de señales: motivación y ventajas.....	68
2.2.1.3	Implementación práctica de la conversión A / D.....	71
2.2.2	La cuantificación.....	73
2.2.2.1	Introducción	73
2.2.2.2	Tipos de cuantificación.....	75
2.2.2.3	El ruido de cuantificación.....	78
2.2.3	Sobremuestreo.....	83
2.2.3.1	Interpolación	83
2.2.3.2	Efecto del sobremuestreo sobre la SNR	84
2.2.3.3	Aplicación del sobremuestreo.....	85

2.2.4	Dither	88
2.2.4.1	El efecto del dither en la audición	88
2.2.4.2	Estudio experimental del dither	89
2.2.4.3	Estudio analítico del dither	90
2.2.4.4	Resumen	98
2.2.5	Noise – Shaping.....	99
2.2.5.1	Introducción	99
2.2.5.2	Estabilidad y ganancia	107
2.2.5.3	Noise-Shaping basado en modelos psicoacústicos	111
3	Tecnologías de desarrollo	119
3.1	El entorno web	119
3.1.1	Introducción.....	119
3.1.2	Lenguajes de presentación	121
3.1.2.1	XHTML.....	121
3.1.2.2	CSS	123
3.1.2.3	Javascript.....	126
3.1.3	Librería de Javascript JQuery	127
3.1.4	Cuestiones de diseño web.....	128
3.1.4.1	Desventajas y puntos débiles del entorno web	128
3.1.4.2	Navegadores de Internet	129
3.1.4.3	Diseño web y usabilidad	131
3.1.5	Análisis y seguimiento del sitio web: Google Analytics.....	137
3.2	Lenguaje de desarrollo Java	141
3.2.1	Características	142
3.2.1.1	Orientación a objetos.....	142
3.2.1.2	Programación de alto nivel.....	144
3.2.1.3	Multiplataforma	145
3.2.1.4	Capacidad de desarrollo de aplicaciones para entorno web.....	145
3.2.1.5	Desventajas	146
3.2.1.6	Conclusiones y notas	147
3.2.2	Módulo de sonido JavaSound	148
3.2.3	Librería de gráficas JFreeChart	150
3.2.3.1	Introducción	150

3.2.3.2	Integración en las aplicaciones del tutorial.....	151
3.2.3.3	Estudio del rendimiento	151
3.2.3.4	Uso de las gráficas.....	154
4	Diseño e implementación del tutorial.....	159
4.1	Diseño web del tutorial	159
4.1.1	Vistas de las áreas del tutorial	163
4.1.2	Esquema de organización del sitio web.....	165
4.2	Aplicaciones web.....	175
4.2.1	Descripción y funcionamiento de las applets.....	177
4.2.1.1	Disposición gráfica de los bloques principales	177
4.2.1.2	Elementos y controles de las applets.....	179
4.2.2	Vista completa de las applets	188
4.2.3	Uso aplicado de las applets	190
4.2.3.1	Applet de cuantificación.....	190
4.2.3.2	Applet de sobremuestreo	191
4.2.3.3	Applet de dither	193
4.2.3.4	Applet de Noise – Shaping (sistema completo)	196
4.3	Desarrollo de la implementación en Java las aplicaciones.....	201
4.3.1	Estructura básica	201
4.3.2	Clases lanzadoras o de arranque	203
4.3.2.1	Proceso básico de ejecución.....	204
4.3.2.2	Métodos y clases internas	206
4.3.3	Clases auxiliares	210
4.3.3.1	ArchivoAudio.....	210
4.3.3.2	Conversor	213
4.3.3.3	Sonido	215
4.3.3.4	MétodosAuxiliares	216
4.3.3.5	Gráficas.....	218
4.3.3.6	Generador Aleatorio	223
5	Conclusiones y trabajo futuro	227
5.1	Conclusiones.....	227
5.2	Trabajo futuro.....	230
	ANEXO I: Distribución y despliegue del proyecto web	233

ANEXO II: Estructura de clases y métodos de las applets.....	237
Índice de figuras.....	243
Pliego de condiciones	247
Presupuesto	251

I

INTRODUCCIÓN

1 Introducción

1.1 Presentación del proyecto. Motivación y objetivos

El proyecto “Tutorial Web de Técnicas de Digitalización de Audio” nace con el objetivo de crear una plataforma de aprendizaje para apoyar el estudio de las técnicas de digitalización de audio a través de la presentación teórica de las mismas y la experimentación práctica.

En concreto, estas técnicas son: el sobremuestreo, el *dither* y el filtrado *Noise-Shaping*, todas ellas, encuadradas dentro del proceso de cuantificación, más genéricamente, la conversión analógica-digital y digital-analógica de señales.

Un requisito fundamental es que será creado para ser utilizado con medios informáticos en red, aprovechando las últimas tecnologías de creación de aplicaciones y acceso a contenidos a través de Internet.

Se enmarca dentro del ámbito académico universitario, en la asignatura “Tratamiento Digital de Audio”, la cual se imparte en la titulación Ingeniería de Sistemas Audiovisuales (equivalente a la antigua Ingeniería Técnica de Sonido e Imagen), aunque no obstante, tiene vocación de convertirse en una referencia para el estudio y profundización en la materia para cualquier estudiante que se interese por ella.

Las técnicas de digitalización de audio son una materia de estudio muy importante dentro de esta especialidad, uno de los conocimientos fundamentales que tradicionalmente el ingeniero de sonido debe dominar en profundidad para comprender el funcionamiento de los sistemas de audio digitales que en el futuro manejará, para aplicarlos en su actividad profesional de manera correcta y útil, tanto si se dedica al diseño de sistemas y equipos de audio como al montaje y puesta en funcionamiento de salas profesionales como estudios de grabación.

El proyecto nace en la Universidad Carlos III como resultado de comprobar la importancia que tiene la práctica en la adquisición de conocimiento y su comprensión por parte del alumno. El objetivo, al fin, es potenciar la experiencia de los alumnos y superar las limitaciones que se tenían con los métodos antiguos de

prácticas, como simulaciones en programas de computación, que si bien, eran efectivos y útiles, no resultaban atractivos y tenían notables carencias, como su escasa integración con la teoría y su poca interactividad.

Es conveniente señalar que la materia tratada puede resultar árida y poco atractiva para el estudiante que se acerca a ella por primera vez, con una aplicación práctica en el mundo del sonido que no se ve de forma inmediata. Sin embargo, ciertamente, las técnicas de digitalización de audio son un pilar fundamental de la ingeniería de sonido, como se ha comentado anteriormente. Por ello, el reto principal de este proyecto es crear una aplicación multimedia que consiga exponerlas de manera sencilla y clara, incidiendo especialmente en su aplicación práctica y que permita realizar simulaciones de forma rápida y eficaz, con una interfaz atractiva, que facilite la experimentación, evitando tediosas tareas de programación y el seguimiento de guías por parte del alumno. Esta manera de plantear el estudio se ha demostrado que es la que da mejores resultados, teniendo en cuenta, que además, en la formación de un ingeniero es esencial la experimentación práctica, más allá de la memorización de teoremas y demostraciones matemáticas sin ver su utilidad práctica.

En cuanto a la materia de estudio en sí, cabe decir que las técnicas estudiadas en este proyecto están bien asentadas y son conocidas y utilizadas desde hace décadas, lo que implica que se parte de una sólida explicación y demostración matemática de las mismas. La digitalización de señales comenzó a ser estudiada a principios del siglo XX con los avances en la Teoría de la Información por parte de Shannon o Nyquist, lo que posteriormente, derivaría en la revolución de las telecomunicaciones que posibilitó la creación de Internet, la telefonía móvil...etc. En el campo del audio, uno de los hitos más destacables es la aparición del soporte CD y los samplers en los años ochenta, tecnologías en las que la aplicación práctica de los conceptos de muestreo y cuantificación de señales es esencial.

El sobremuestreo fue propuesto por los ingenieros de Sony que desarrollaron el Compact Disc como una solución para mejorar su funcionamiento y solventar ciertas dificultades técnicas; el *dither* es una técnica conocida desde el momento en el que al pasar de los sistemas analógicos a los digitales se constató el

problema de la distorsión; el *Noise-Shaping* fue propuesto por Cutler en 1956 para reducir el ruido de cuantificación y su principio básico de funcionamiento sigue teniendo vigencia a día de hoy. No obstante, todas ellas siguen siendo objeto de investigación y existen abiertas interesantes vías de exploración como el Noise – Shaping basado en modelos psicoacústicos. Por tanto, el estudio de estas técnicas no se limitará a la exposición de la teoría asentada, si no que se presentará una visión básica de las principales líneas de investigación que se siguen actualmente para que resulte un tutorial moderno y actualizado.

El proyecto presenta varias cuestiones que resolver de muy diferentes ámbitos. Por una parte, tiene una notable exigencia en cuanto al estudio teórico de estas técnicas, es decir, se deberá elaborar una presentación teórica de las mismas e investigar los últimos avances y propuestas sobre ellas, antes de pasar a la elaboración del material didáctico. No sólo eso, es más, se plantea un trabajo de clasificación y evaluación de los contenidos esencial, no se trata simplemente de transmitir la teoría científica de forma directa e inmediata, tal y como aparece en las fuentes de referencia, sino que hay que procesarla para presentarla al alumno de forma adecuada, de modo que aprenda lo más importante, lo asimile, y se despierte en él interés por profundizar en la materia, mostrándole el camino para estudiar temas más avanzados.

Por otro lado, se requerirá la creación de una herramienta de simulación que conlleve la producción de aplicaciones informáticas de cierta complejidad, con el consiguiente proceso de planificación y codificación de software.

En conclusión, se trata de un proyecto multidisciplinar, pero que no obstante, se puede inscribir dentro de la ingeniería audiovisual, en concreto, en las áreas referentes a la creación de contenidos multimedia por una parte y por otra, el estudio de la teoría sobre la digitalización de señales.

1.2 Tele-educación y e-learning

La tele-educación, o tele-enseñanza, como también es conocida, es un concepto muy amplio, engloba toda una serie de sistemas y métodos de apoyo al aprendizaje que se valen de los últimos avances tecnológicos en el campo de la informática y las telecomunicaciones para potenciar el desarrollo de cursos a distancia, en los cuales, profesor y alumno no se encuentran próximos físicamente, o bien, completar con material adicional cursos impartidos con los métodos tradicionales de enseñanza presencial.

Dada la propia naturaleza de estos sistemas, muy heterogéneos y variados, que resuelven problemas muy dispares, existen diferentes enfoques o puntos de vista a la hora de definir este campo y clasificar los diferentes elementos que lo forman.

Se puede tomar, por ejemplo, una definición dada por el Ministerio de Obras Públicas y Transportes en 1996:

"Se entiende por tele-educación al desarrollo del proceso de formación (reglada o no reglada), basado en el uso de tecnologías de la información y las telecomunicaciones, que posibilitan la realización de un aprendizaje interactivo, flexible y accesible a cualquier receptor potencial. Modificando el ritmo, la frecuencia, y las coordenadas espacio-temporales favoreciendo una educación más individual y flexible."

Esta definición incide en las ideas fundamentales comentadas anteriormente, el hecho de aprovechar la tecnología para potenciar la educación dotando al proceso formativo de nuevos aspectos hasta entonces irrealizables sin las telecomunicaciones, como son la interactividad, la flexibilidad y la ubicuidad.

Hasta finales del siglo XX pueden destacarse, como sistemas mas importantes que han formado parte de procesos de tele-enseñanza, la utilización de vídeos y cassetes desde los ochenta, emisiones por satélite y la videoconferencia a partir de principios de los noventa, y la introducción de la informática personal, gracias a la irrupción del PC a mediados de los noventa, que trajo consigo la creación de software y aplicaciones multimedia especialmente diseñadas para el

aprendizaje. Por tanto, la definición de MOPTMA expuesta anteriormente, se circunscribe en este contexto, de la era “pre-Internet”, en la que la tecnología empezaba a asentarse en el mundo de la educación, pero sus posibilidades eran todavía limitadas, aunque las bases y los objetivos ya se habían establecido y se vislumbraba un gran potencial. [41].

En realidad, la idea de la tele-educación surge mucho antes, a la par que la aparición de las propias telecomunicaciones, pues ciertamente, es un concepto sumamente atractivo y que abre muchas posibilidades. La radio y el cinematógrafo, simplemente, ya eran medios con un enorme potencial para la difusión de cursos y material didáctico a una gran masa de gente. Por eso, ya 1922, Edison se aventuraba a asegurar que las telecomunicaciones revolucionarían el sistema educativo y harían desaparecer los libros de texto. A partir de entonces, las experiencias en este sentido han sido numerosas, aunque no han supuesto una revolución, si no un complemento, sólo anecdótico, al sistema de enseñanza tradicional. Las premisas de interactividad y flexibilidad se cumplían de forma muy limitada, pues al fin y al cabo, se trataba de sistemas broadcast con poca capacidad de personalización.

La popularización de la red Internet, a principios de siglo XXI, supuso una revolución a todos los niveles en la sociedad, y en la tele-educación, en concreto, fue la clave para la innovación y el desarrollo de sistemas que marcaron la diferencia respecto a todo lo anterior, pues ahora sí, los conceptos de interactividad y flexibilidad se plasmaban de forma efectiva y palpable.

Esto dio lugar al concepto de *e-learning*, que se refiere a los sistemas de tele-educación que aprovechan, específicamente, las posibilidades de Internet y la informática personal a la hora ofrecer su material y vías de comunicación. Marc Rosenberg, experto de reconocido prestigio en el ámbito de la tele-educación, definía, exactamente, *e-learning*, de la siguiente manera:

“El uso de tecnologías Internet para la entrega de un amplio rango de soluciones que mejoran el conocimiento y el rendimiento.”

Además, completó esta definición con tres principios básicos que un sistema de *e-learning* debe cumplir:

1. Los sistemas de *e-learning* son accesibles a través de la Red, por lo que la información puede ser actualizada, almacenada, recuperada y distribuida de forma inmediata, además de compartida por un gran número de personas.
2. El acceso a la Red se realiza mediante ordenadores personales conectados a Internet
3. El aprendizaje se enfoca desde un punto de vista mucho más amplio que los sistemas tradicionales de instrucción formativa.

Otros autores, en cambio, definen *e-learning* desde una perspectiva más amplia, e incluyen en él cualquier tipo de sistema de aprendizaje que utilice medios electrónicos, en red o no, para enriquecer el proceso de enseñanza. En todo caso, las definiciones formales son secundarias para los objetivos del proyecto, pues como se puede ver fácilmente, el concepto de tele-educación es muy subjetivo por lo que es proclive a que surjan discrepancias. Lo fundamental es analizar hasta que punto se ha progresado, en la práctica, en la implementación de sistemas de tele-enseñanza.

El progreso en el campo de la tele-educación ha dado lugar a productos y sistemas muy complejos, que abarcan un gran número de funciones. Formalmente conocidos como LMS en sus siglas en inglés (en español, Sistema de Gestión de Tele-enseñanza), estos productos ofrecen todos los servicios necesarios para proveer un curso a distancia, como son la gestión y distribución del material didáctico, métodos para facilitar al alumno su autoevaluación, sistemas de evaluación no presencial y finalmente, vías de comunicación e interacción entre alumno y profesor [37].

Sin embargo, estos sistemas quedan fuera totalmente del propósito de este proyecto, pues el objetivo no es crear una plataforma que abarque un gran número de funciones, con herramientas de comunicación como videoconferencia, foros, etc, para posibilitar la realización de un curso on-line. Las pretensiones son mucho más

humildes, siendo la meta principal crear una herramienta para apoyar la impartición presencial de una asignatura universitaria, aunque se diseñe de modo que sea factible su utilización fuera del ámbito de la asignatura, como material de referencia para personas que quieran ampliar sus conocimientos.

En este sentido, se pueden citar las ideas que Rosenberg expone en sus últimos trabajos, en los que plantea que debe superarse el concepto tradicional de tele-educación centrado en exclusiva en cursos on-line con una estructura muy rígida y anquilosada, por otro mas amplio en el que el foco sea el conocimiento. Es decir, poner a disposición del usuario una serie de herramientas que puedan ser utilizadas de muy diversa manera: para introducirse en una materia, como consulta o referencia, etc.

Por ello, la aplicación que se pretende construir en este proyecto se plantea como un conjunto formado por una exposición teórica más una serie de herramientas prácticas, que puede ser utilizado de muy diversa manera según las necesidades del usuario. Podrán consultarse los conceptos básicos si el usuario desconoce la teoría y quiere empezar de cero, ampliar sus conocimientos si se utiliza como complemento a la asistencia a clase en la universidad, o simplemente, ignorar la teoría, si ya se conoce, y acceder directamente a las aplicaciones prácticas para realizar experimentos.

El ambiente en el que esta aplicación será utilizada, también, se puede considerar indeterminado. Podrá ser utilizada por un alumno desde su casa conectándose a Internet, en las aulas de la universidad, o incluso, en una clase equipada con proyector y equipo de audio y manejada por el profesor. Por eso, el planteamiento de esta aplicación se acerca mas a la idea de enseñanza asistida por recursos electrónicos, independientemente del lugar y el contexto, ya que el profesor puede estar o no presente, que al concepto de enseñanza a distancia.

Un nivel superior, que Rosenberg describe como "*e-learning* centrado en el conocimiento", es la organización de la información y el material didáctico en repositorios a los que el usuario puede acceder libremente, cuando lo necesite, para obtener datos, explorar un tema, mediante búsquedas o a través de directorios. Por ello, aunque queda fuera del propósito de este proyecto, es

necesario diseñarlo con la mente puesta en que posteriormente pueda ser integrado como un componente dentro de una aplicación más general, que englobe toda la materia de la asignatura Tratamiento Digital de Audio, o mas ambicioso aún, que aborde el estudio del Procesamiento de la Señal y las Comunicaciones, desde un punto de vista teórico y práctico. [38]

Todo lo expuesto hasta ahora sirve para enmarcar el proyecto dentro del contexto actual de la tele-educación, de un modo muy general. Es necesario profundizar, más concretamente, en cómo se construyen las aplicaciones del tipo que se desarrollará, es decir, un tipo de aplicación sencilla, que tiene como objetivo experimentar con ciertos algoritmos, implementada en Java como applet disponible dentro de una página web.

La experiencia adquirida en la Facultad de Informática de la Universidad Politécnica de Madrid en el desarrollo de aplicaciones web de tele-enseñanza puede ser tomada en consideración para extraer algunas ideas básicas de cómo plantear estas herramientas [39]. Este centro universitario ha desarrollado un conjunto de applets que sirven de práctica en el aprendizaje de los algoritmos de grafos, como el de Dijkstra o los mapas de Karnaugh, que son esenciales en campo de la computación y la telemática. La materia tratada está muy alejada de este proyecto, no tiene nada que ver con las técnicas de digitalización de audio ni con el tratamiento de señal, sin embargo, en la problemática de su enseñanza tienen muchos puntos en común, pues ambas son técnicas difíciles de entender en un aula tradicional, sobre una pizarra. Tras la exposición teórica, es imprescindible que el alumno compruebe, interactivamente, de forma visual, y en nuestro caso, además, auditiva, el funcionamiento y su efecto.

A partir de su experiencia en la utilización de esta herramienta, se han podido extraer algunas conclusiones respecto a que cualidades deben tener las aplicaciones web para ser efectivas pedagógicamente. Hay que insistir, que no obstante, estas conclusiones se aplican específicamente a las applets dedicadas a los algoritmos de grafos, pero por lo expuesto anteriormente, parte de ellas pueden ser valoradas para este proyecto.

Dichas ideas son:

- Interactividad: las applets esperan a la respuesta del usuario y la evalúan.
- Flexibilidad: las applets admiten el planteamiento de cualquier problema, con cualquier tipo de parámetros.
- Simplicidad de uso
- Claridad en la representación gráfica de los resultados
- Visualización de la estructura del algoritmo
- Disponibilidad para el alumno, en cualquier momento y lugar
- Peso de descarga reducido

1.3 Estado del arte

La materia tratada por el proyecto es muy concreta en cuanto a los contenidos didácticos que se impartirán con el tutorial web, pero sin embargo, muy amplia en cuanto al hecho que supone plantear la creación de una herramienta de apoyo al aprendizaje, es decir, una utilidad de tele-educación. Por tanto, el estado del arte puede abordarse desde perspectivas muy diferentes, desde un punto de vista global sobre el estado de la tele-educación, hasta el ámbito más concreto del audio y las aplicaciones relacionadas con él. Hay que tener en cuenta que muchos de los aspectos tratados por la enseñanza a distancia no competen a este proyecto, como son todos los relacionados con la comunicación entre profesor y alumno, métodos de evaluación no presencial, etc, ya que la finalidad única y exclusiva de este proyecto es crear una aplicación de audio sencilla que ayude a comprender el funcionamiento de las técnicas de digitalización de audio, acompañadas de una exposición de los fundamentos teóricos que será, hay que reiterar, un complemento al material del profesor, no el contenido principal de la asignatura. Las cuestiones tecnológicas y en concreto, en el ámbito del software, en cambio, sí que juegan un papel decisivo, pues al fin y al cabo, el objetivo es presentar una aplicación innovadora, de carácter didáctico, que aporte algo más respecto a lo que había antes, en concreto, en el ámbito del audio.

¿Qué tecnologías son las más utilizadas para la creación de aplicaciones de audio online?, ¿Existe ya algún tipo de aplicación que aborde la temática estudiada en este proyecto? ¿Cuáles son las principales fortalezas y carencias del material creado hasta ahora? Éstas son las principales preguntas que se plantean a la hora de abordar el estado del arte. Por tanto, se presentarán una serie de proyectos de aplicaciones de audio, haciendo especial hincapié en las relacionadas directamente con las técnicas de digitalización de audio en cuanto a la temática y con el mundo de Java y las applets como escenario.

Aplicación "Audio Effect Applets for Music Processing"

La aplicación "Audio Effect Applets for Music Processing", creada por Mijail Guillermand, Christian Ruwee y Udo Zolzer, en la universidad alemana Helmut-Schmidt, de forma casi paralela al planteamiento de este proyecto, puede tomarse

como punto de referencia fundamental, pues se trata de una aplicación de carácter didáctico, implementada como un applet Java disponible online, de acceso gratuito y código libre (GNU), que aborda los mismos temas relacionados con el campo del audio aquí tratados. Puede decirse que es el proyecto más importante y avanzado presentado hasta ahora en este campo, con lo cual, es esencial hacer un estudio previo del mismo, que puede llevar a encontrar posibles mejoras o un nuevo enfoque [40].

Este proyecto consiste en un conjunto de applets dedicadas al tratamiento de audio, que mediante la simulación de ciertos procesos y técnicas, permiten comprobar el efecto auditivo y sobre el espectro de la señal de un modo práctico e inmediato. Las applets tratan temas muy variados, se pueden citar algunos tales como el filtrado de señal, efectos de retardo, ecualizador de octavas, convolución y finalmente, una dedicada a la cuantificación, *dither*, sobremuestreo y *Noise-Shaping*. Como es obvio, esta última es la más interesante por coincidir en temática con la del proyecto, y por tanto, se efectuará un breve análisis sobre ella.

El applet de técnicas de digitalización de audio ofrece como resultado una representación dinámica, en tiempo y frecuencia, de la señal de audio, acompañada de la reproducción inmediata y continua de ésta. Aporta un par de ejemplos de señales predefinidas, aunque permite la carga de ficheros seleccionados por el usuario. Como datos de entrada, permite la configuración de una serie de parámetros relacionados con las técnicas tratadas, como son, el número de bits de cuantificación empleado, el tipo de *dither* (triangular, uniforme, paso alto), el factor de sobremuestreo y finalmente, la aplicación o no de un filtro de Noise-Shaping de varios órdenes. Acompañando la configuración de los módulos realizada por el usuario, se presenta un diagrama de bloques del sistema que se modifica dinámicamente.

A continuación, se presentan una serie de casos de uso donde se puede observar el funcionamiento básico de esta aplicación.

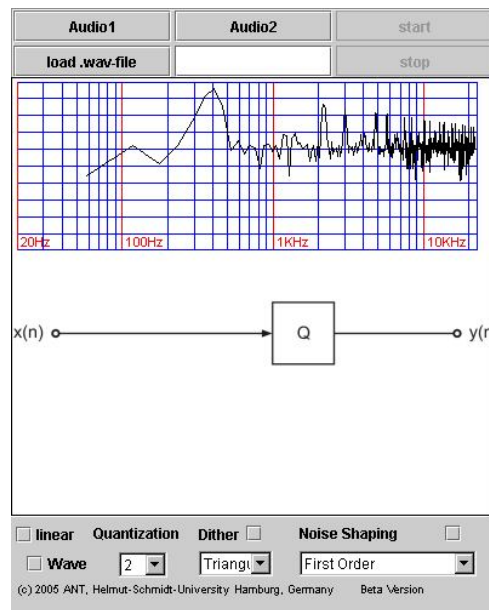


Figura 1-1 Audio Effect Applets for Music Processing - Cuantificación

En primer lugar, se muestra una gráfica del efecto de cuantificar la señal sin aplicar ningún tipo de técnica. Al tomarse como ejemplo un tono, se puede observar fácilmente el ruido como la parte de señal fuera de su frecuencia. Si el número de bits empleado disminuye, se observa, intuitivamente, como el ruido aumenta.

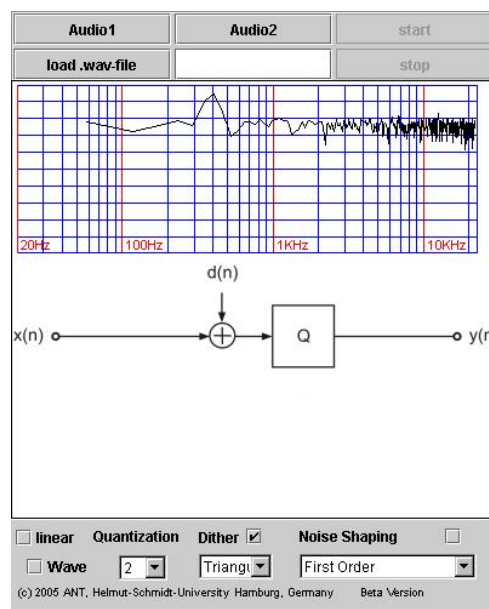


Figura 1-2 Audio Effect Applets for Music Processing - Dither

Luego, se puede introducir *dither*, con lo que se aprecia con claridad como el ruido se vuelve uniforme y blanco, desapareciendo la distorsión, que son las

réplicas de señal que aparecían como armónicos del tono por el hecho de cuantificarlos.

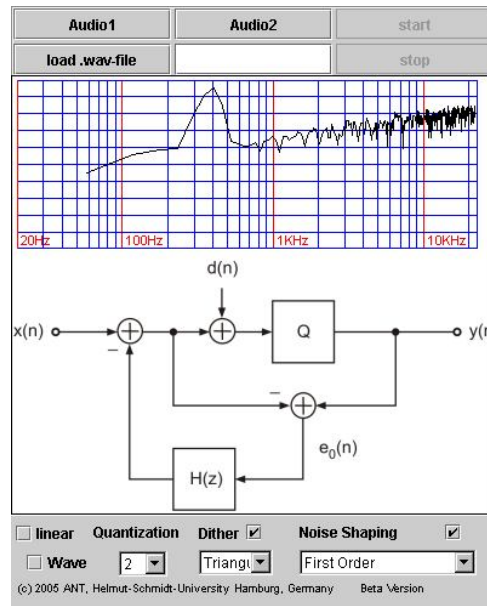


Figura 1-3 Audio Effect Applets for Music Processing – Noise Shaping

Posteriormente, se añade el módulo de Noise-Shaping, observándose fácilmente una modificación en el espectro del ruido de cuantificación, que se desplaza a las altas frecuencias.

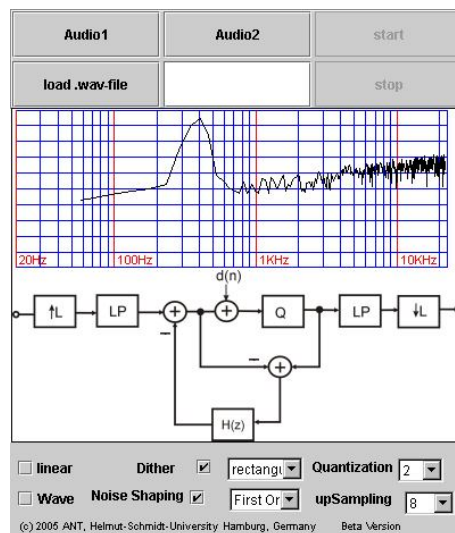


Figura 1-4 Audio Effect Applets for Music Processing - Oversampling x 8

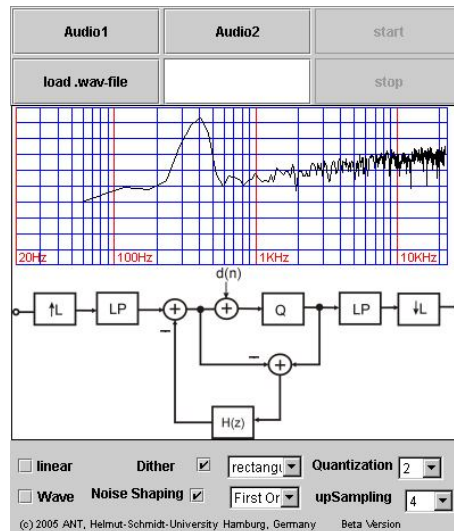


Figura 1-5 Audio Effect Applets for Music Processing - Oversampling x 4

El applet, finalmente, permite activar la configuración del módulo de sobremuestreo, lo que permite al usuario seleccionar entre una serie de factores de multiplicación, observando en tiempo real su efecto.

Esta aplicación es una buena herramienta para observar de una manera sencilla y rápida el efecto de las técnicas de digitalización de audio. Destaca por su facilidad de uso, con apenas un par de “clicks” de ratón se puede ver de manera inmediata cómo el ruido de cuantificación aumenta o disminuye en función del valor de los parámetros de configuración de estas técnicas. Su funcionamiento es robusto y su interfaz simple y manejable. Además, ofrece la posibilidad al usuario de cargar una señal de audio propia, aparte del tono y una señal musical predeterminadas. En definitiva, consigue con eficacia su propósito, que es mostrar el resultado de aplicar técnicas de digitalización de audio de forma simple y fácil.

Tomando esta aplicación como referencia puede reflexionarse sobre las posibilidades que se abren una vez se ha visto la efectividad que tiene una herramienta para observar las técnicas de digitalización de audio. Aunque, como se ha comentado, la aplicación de Mijail Guillermand, Christian Ruwee y Udo Zolzer cumple a la perfección con su propósito, puede aportarse una serie de ideas para extender su funcionalidad o incluso, replantear su diseño con el objetivo de crear una herramienta más completa.

Su sencillez es una virtud, pero a la vez, puede verse como una desventaja. Como fortaleza se puede resaltar que esto permite al alumno acercarse de forma

fácil a las técnicas de digitalización y despertar su interés por ellas, al observar los resultados de forma muy rápida y visual. Sin embargo, el applet tiene unas capacidades muy limitadas en cuanto a la posibilidad de realizar un estudio sistemático y más profundo de las técnicas, es decir, una vez se adquiere una impresión inicial sobre ellas, poco más se puede hacer que “jugar” con el applet. Las opciones para inspeccionar el espectro en detalle, de observar por separado señal y ruido, de tomar datos numéricos de potencia de ruido para realizar un análisis cuantitativo y no sólo cualitativo, son inexistentes. Además, hay que añadir, que el applet se centra tan sólo en la observación del resultado final, presentando las técnicas de digitalización como una “caja negra”, no ofrece la posibilidad de inspeccionar, de un modo didáctico, la manipulación que sufre la señal a lo largo de las diferentes etapas que las componen.

Por todo ello, se llega a la conclusión que merece la pena aportar un nuevo planteamiento en el estudio práctico de las técnicas de digitalización de audio, cuyo objetivo no tiene porqué ser sustituir a esta aplicación, si no complementarla.

Las ideas esenciales que servirán de guía para conseguir un nuevo producto original e innovador son las siguientes:

Obtener una implementación en Java igual de efectiva de las técnicas de digitalización de audio, aunque replanteada desde cero para investigar su posible mejora que además, este modularizada en componentes fácilmente integrables en nuevas aplicaciones.

Desarrollar una interfaz más amplia, que incorpore nuevas funcionalidades. La representación de la señal en tiempo y frecuencia es el componente con más posibilidades de mejora. Sería deseable aportar una herramienta que permita seleccionar el instante temporal donde observar el espectro y no sólo mirar su evolución a la par que se reproduce la señal. Además, sería conveniente completarlo con una representación, en promedio, para toda la señal. Finalmente, añadir alguna utilidad que facilite el uso de las gráficas, como un zoom o flechas de desplazamiento para recorrer la señal en el tiempo.

Presentar una representación de la señal y el ruido por separado, y diferentes capturas en puntos de interés del sistema, no solo de la salida. Además,

presentar gráficas comparativas en las que se superpongan la señal a la que se ha aplicado la técnica y otra sin modificar, para observar, con mayor claridad, el efecto.

Aportar una serie de resultados numéricos, como medidas de la potencia del ruido de cuantificación, para realizar un estudio cuantitativo de los efectos, más allá de la simple impresión auditiva, en la señal reproducida, o visual, en las gráficas de espectro.

Buscar una descripción más exhaustiva de las técnicas de digitalización, en vez de centrar la atención sólo en los resultados. Se puede tomar como ejemplo el sobremuestreo, sería interesante comparar sobre una misma gráfica la señal muestreada a la frecuencia original con la señal sobremuestreada, de modo que se vea como al extenderse el ruido en el espectro, pero no la señal, se reduce el nivel del primero. En el caso del *dither*, también, sería una buena idea observar con mayor precisión como se transforman los armónicos en ruido uniforme y como en el tiempo, esto se traduce en una aleatorización de los pulsos rectangulares provocados por la cuantificación, dentro de los cuales, aparecen basculamientos que reducen la distorsión.

Aplicación "Virtual Laboratories"

El producto de software "Virtual Laboratories", desarrollado por Graham Goodwin, en la Universidad de Newcastle, Australia, consiste en un conjunto de aplicaciones de tele-educación diseñadas como herramienta de apoyo al aprendizaje a nivel universitario. Se basan en simular una serie de experimentos bien conocidos en diferentes materias como física, electricidad, sistemas, automática y comunicaciones, y que se suelen realizar en los laboratorios de los primeros cursos de las enseñanzas técnicas.

De este producto es interesante mencionar, en concreto, por su relación con este proyecto, la aplicación "Audio Signal Processing", que consiste en una herramienta de simulación de Noise-Shaping, centrada en dos temas principales, el análisis de proceso de cuantificación de señales de audio por un lado, y por otro, el estudio de la sensibilidad del oído humano, es decir, las curvas de nivel, y como se puede aplicar a la mejora del proceso de cuantificación.

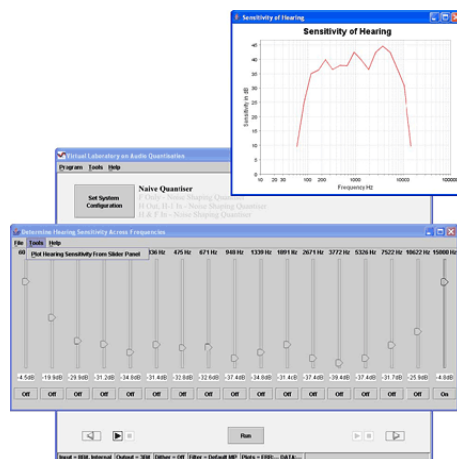


Figura 1-6 Virtual Laboratories

Esta aplicación, más que por su planteamiento o características técnicas, es útil mencionarla porque su modelo de distribución y oferta al público es justo el opuesto a la presentada anteriormente. Se trata de un producto que requiere licencia para ser utilizado, es decir, es de pago, y no se encuentra online en la Red, sino que debe ser encargado para recibir un instalable en el ordenador, lo que conlleva un esfuerzo de instalación y puesta en marcha. Esto pone de manifiesto que cada vez es mayor la necesidad de utilizar herramientas de apoyo al aprendizaje en la universidad, que aborden los problemas desde un punto de vista práctico, además de la bibliografía recomendada, hasta tal punto, que se ha convertido en una posibilidad de negocio.

La primera aplicación presentada ofrece una buena calidad y además, teniendo en cuenta que es gratuita y libre, se puede considerar una opción muy recomendable. La segunda aplicación no puede ser valorada en cuanto a su calidad ya que no es posible probarla, simplemente, hay que fiarse de su publicidad. Es menester de valorar en cada caso cuando es suficiente con utilizar un producto como el primero o merece la pena pagar por tener acceso a un software, teóricamente, mas profesional.

De cara al proyecto aquí tratado, conviene reforzar la idea de que merece la pena seguir investigando y avanzando en la creación y mejora de aplicaciones de libre distribución como apoyo al aprendizaje en el ámbito universitario, como la de Mijail Guillermand, Christian Ruwee y Udo Zolzer, que son claramente, la opción de futuro.



CONCEPTOS TEÓRICOS

2 Conceptos teóricos

2.1 Procesado digital de señal

El procesado digital de la señal engloba una serie de procedimientos y técnicas de muy diversa índole que se aplican en el tratamiento de señales y que intervienen en los sistemas digitales que se construirán para implementar las técnicas tratadas en el proyecto.

El objetivo de este apartado es dar una visión general de estas técnicas, con una breve descripción teórica de las mismas para exponer su utilidad y comprender su funcionamiento de forma genérica, y además, dar ciertas indicaciones de cuál será su papel en los sistemas del proyecto en los que intervendrán, justificando las opciones elegidas en cada caso.

Es decir, se pretende realizar una descripción general de dichas técnicas para luego entrar en la materia propia del proyecto en los capítulos sucesivos y de esta manera, comprender mejor la presencia de ciertos bloques y su funcionamiento en los sistemas de digitalización de audio que se construirán. La materia tratada es compleja y tiene una gran profundidad, no se trata de conceptos triviales, por tanto, conviene aclarar que el objetivo del proyecto no es hacer previamente un estudio teórico, de carácter exhaustivo, de dichas cuestiones, si no, simplemente, dar una visión general de ellas y en particular, de sus implicaciones en el proyecto.

Entrando en materia, se puede tomar como primer tema de interés el análisis espectral de la señal. Es uno de los pilares fundamentales del tratamiento de audio y en este proyecto tiene un papel central. Esta materia, que se sustenta en la rama de las matemáticas conocida como “análisis armónico de señales”, comúnmente conocida por la Transformada de Fourier, es una herramienta básica para comprender el funcionamiento de las técnicas de digitalización de audio. Su utilidad reside en que permite observar la distribución frecuencial de una señal, y por tanto, ver las modificaciones sobre el espectro que aplican estas técnicas. Relacionados directamente con el análisis espectral de señales, surgen otros temas auxiliares, como la FFT (Fast Fourier Transform), el inventariado de señal y la

estimación del espectro a través del método de Welch. Estas cuestiones tratan, realmente, de la implementación práctica con la que se llevará a cabo el análisis espectral.

El filtrado de señal es otra materia de gran importancia y que interviene en los sistemas contruidos para implementar las técnicas de digitalización de audio. Su objetivo es realizar modificaciones sobre el espectro de las señales, para amplificar o atenuar ciertas bandas de frecuencias. Tiene múltiples aplicaciones y en concreto, en el campo del audio, es un bloque esencial de la mayoría de los sistemas. Por ejemplo, son ampliamente conocidos los filtros paso bajo, paso alto y paso banda y aún más, da lugar a aplicaciones complejas como los ecualizadores, aunque en este proyecto, tan sólo se requiere la implementación de algunos filtros básicos necesarios para el funcionamiento de las técnicas de digitalización. Tienen su origen en los sistemas analógicos, para, posteriormente, haberse definido su implementación en el dominio digital.

2.1.1 Análisis espectral de señal

El análisis espectral de una señal temporal consiste en obtener su distribución en el dominio de la frecuencia, es decir, se trata de descomponerla en las frecuencias que la forman, dentro de un rango dado, obteniendo, además, que peso tiene cada una de ellas en el conjunto. Las señales pueden tener naturaleza muy diferente: radioeléctricas, de sonido, etc, pero en definitiva, todas en ellas son fenómenos ondulatorios que pueden modelarse como una señal en el tiempo compuesta por la superposición de ondas con un determinado período y amplitud, la señal se puede describir completamente, por tanto, con esta información. El espectro de una señal es una información de gran utilidad que da una idea del comportamiento y características de la misma. Por otro lado, desde el punto de vista opuesto, mediante la representación de una señal en el dominio de la frecuencia y además, con la información de la fase, se puede realizar la transformación inversa, es decir, obtener la señal en el tiempo.

El tiempo y la frecuencia, en conclusión, son dominios opuestos en los que puede observarse una señal. La inspección del espectro de una señal, la frecuencia, aporta una información muy valiosa, de gran importancia para evaluar la acción de

las técnicas de digitalización de audio. Lo habitual es disponer de una representación en forma de onda de la señal, es decir, en tiempo y el objetivo será obtener el espectro, para lo cual, hay que recurrir al citado análisis armónico de señal o análisis de Fourier, como también se conoce.

En el ámbito del sonido, el espectro de la señal determina en buena medida la sensación auditiva que se produce en el oído. Las notas musicales, por ejemplo, son un conjunto de tonos, deltas en el espectro, distribuidos de una determinada manera, es decir, siguiendo un cierto orden, lo que las diferencia unas de otras y del instrumento del que proceden. Las señales sonoras que tienen la mayor parte de la potencia situada en las bajas frecuencias, son percibidas como graves, y al contrario, las que tienen la mayor parte de su potencia situada en las altas frecuencias, son percibidas como agudas. Por tanto, al inspeccionar la forma del espectro de una señal, se puede tomar una idea básica de “cómo es” y comparando el espectro de una señal a la entrada y a la salida de un sistema, o en puntos intermedios de interés, se apreciará la acción del sistema sobre la frecuencia, lo que es una de las características básicas de un sistema. Para el proyecto, por ejemplo, es de gran interés comparar el espectro de la componente de señal original con el de error de cuantificación, y observar como son modificados por los sistemas aplicados, analizando así que resultado provocarán finalmente en la sensación auditiva.

El ruido, que es un elemento clave en este proyecto, para el caso ideal de que se trate de ruido blanco con fdp uniforme, adopta en el espectro la forma de una banda rectangular distribuida por igual en todo el espectro. Uno de los principales objetivos de este proyecto es analizar una fuente de ruido con un origen concreto, el ruido de cuantificación, desde el punto de vista de aproximarle a ruido blanco con fdp uniforme, lo que es, en esencia, realizar un análisis su espectro; por ello, esta herramienta es una de las claves, tanto para explicar y realizar un estudio de la acción y el funcionamiento de las técnicas de digitalización de audio, como para construir las aplicaciones que se servirán de ella para exponer al usuario, de forma práctica, esta materia.

La representación gráfica que se tomará para observar el espectro es la “Densidad Espectral de Potencia”, concepto derivado que se explicará más

adelante; previamente, se explicará la esencia de este análisis que son las series y la transformada de Fourier. Otro concepto esencial a la hora de estudiar el espectro de una señal es el enventanado con el que se calcula éste para un determinado tramo temporal, lo que implica tomar una decisión en cuanto a la resolución en tiempo y la resolución en frecuencia del analizador de espectro, ya que son parámetros opuestos.

No hay que olvidar, no obstante, que la inspección de la señal en el tiempo, es decir, la observación de su forma de onda, también puede tener un gran interés. En el caso del ruido de cuantificación, por ejemplo, al tratarse de un proceso estocástico (idealizado, no en la práctica), es difícil (en realidad, imposible) observar su presencia para un instante temporal concreto, por la esencia misma de los procesos estocásticos, que sólo tienen sentido como un promediado para un tramo de tiempo de la señal, pero aún así, conviene apreciar ciertos fenómenos que se ven con especial claridad en el tiempo. En conclusión, aunque el análisis espectral de la señal es la herramienta básica para el proyecto, el análisis temporal no se debe despreciar totalmente [4] [7] [25].

2.1.1.1 El análisis armónico de señales

El análisis armónico, cabe recordar, consiste en la descomposición de una señal periódica en sus componentes armónicas, es decir, en las frecuencias que la componen, para obtener el espectro. El origen se encuentra en el estudio matemático conocido como "series de Fourier".

Las series de Fourier son una aplicación matemática que permite llevar a cabo la descomposición en frecuencias que se ha explicado. Para ello, se construye la señal dada como una combinación lineal de un número, eventualmente infinito, de ondas periódicas senoidales. Es decir, explicado de un modo coloquial, se trata de obtener la señal sumando un número arbitrario de senos y cosenos multiplicados cada uno de ellos por su peso correspondiente, dependiendo de la presencia del armónico (frecuencia) que representan en la señal. A estos pesos, se les denomina "coeficientes de Fourier" y su cálculo es la clave para llevar a cabo el desarrollo en serie de Fourier .

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} [a_n \cos(nx) + b_n \sin(nx)]$$

a_n y b_n son los citados coeficientes de Fourier que se deben obtener.

La ecuación expuesta suele presentarse sustituyendo la variable x por ωt , con el objetivo de manejar la expresión en función del tiempo y de este modo, ver las componentes que combinadas linealmente forman la señal en el tiempo.

$$f(t) = \sum_{k=-\infty}^{\infty} C_k e^{j\omega k t}$$

Recordando las ecuaciones de Euler:

$$e^{i\omega t} = \cos(\omega) + i \sin(\omega)$$

se puede apreciar fácilmente como el desarrollo en serie de Fourier permite obtener las frecuencias de las que se compone la señal $f(t)$.

Es necesario precisar que este análisis se ha realizado para el caso de señales continuas. Para las señales discretas, existe una versión análoga de estas ecuaciones. Aunque el objetivo del proyecto es obtener el espectro de señales de audio digitalizadas, que por tanto, son discretas, el concepto es el mismo por lo que no es necesario presentarlo de nuevo. No obstante, más adelante, se tratará la implementación práctica para señales discretas en el apartado de la FFT [5].

2.1.1.2 La Transformada de Fourier

La Transformada de Fourier es una herramienta matemática que permite obtener la distribución en frecuencia de una señal aperiódica, a diferencia de las series de Fourier, que se aplican sobre señales periódicas. A partir de una función continua $f(t)$, se obtendrá otra función continua $F(\omega)$ que representa la distribución en frecuencias de dicha señal [5].

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt$$

La Transformada Inversa de Fourier permite recuperar la señal $f(t)$ en el tiempo a partir de su espectro, $F(\omega)$

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{-j\omega t} d\omega$$

La Transformada de Fourier es, en realidad, un caso particular de la Transformada de Laplace con $S = j\omega$, y $\omega = 2\pi f$.

Mediante el uso de la delta de Dirac, se puede emplear, también, la transformada de Fourier para encontrar la distribución en frecuencia de señales periódicas. Esto es así porque, en la práctica, una señal periódica, está compuesta por una o más componentes periódicas, es decir, una serie de ondas senoidales sumadas, de una determinada frecuencia cada una. La delta de Dirac es una función que toma valor infinito en un determinado punto y es cero para el resto, por lo que su integral es 1. En el dominio del espectro, $F(\omega)$, la señal periódica se puede representar como un tren de deltas, cada una de las cuales está centrada en la frecuencia de la componente armónica que representan.

$$x(t)_{\text{periódica}} \xleftrightarrow{F} \sum_{k=-\infty}^{\infty} a_k 2\pi \delta(\omega - k\omega_0)$$

Para el caso de señales discretas, la Transformada de Fourier tiene la siguiente expresión:

$$f_j = \sum_{k=0}^{N-1} x_k e^{-\frac{2\pi i}{N} jk} \text{ para } j = 0, \dots, n-1$$

Siendo su inversa:

$$x_k = \frac{1}{N} \sum_{j=0}^{N-1} f_j e^{\frac{2\pi i}{N} jk}$$

Las fórmulas presentadas exponen el sustento matemático básico para obtener el dominio espectral de una señal. Estas transformadas constan de una serie de propiedades y definiciones que las complementan e indican características relevantes de las mismas, pero no es necesario indicarlás ya que el objetivo esencial es exponer de forma breve la idea fundamental que encierra la transformada de Fourier y el análisis armónico, acompañándolo de las ecuaciones esenciales [5].

2.1.1.3 Densidad Espectral de Potencia y caracterización de procesos estocásticos

La densidad espectral de potencia es una herramienta matemática que permite conocer la distribución espectral de la potencia de una señal, calculándola a partir de su Transformada de Fourier, concepto que se ha expuesto antes. Explicado de una manera más simple, esta función indica cuanta potencia se concentra en cada una de las frecuencias que forman una señal.

La densidad espectral de potencia tiene especial interés desde un punto de vista práctico como medio para caracterizar procesos estocásticos. En el caso del ruido blanco, por ejemplo, su densidad espectral de potencia es una banda uniforme que se distribuye por igual por todo el espectro, es decir, el valor de potencia de todas las frecuencias es el mismo. Para las técnicas de digitalización de audio, en las cuales, la clave se encuentra en modelar el ruido de cuantificación desde el punto de vista de aproximarlos a ruido blanco, por tanto, el análisis de su densidad espectral de potencia es un elemento fundamental para observar su acción sobre la señal y el ruido [7].

La DEP (Densidad Espectral de Potencia) se define matemáticamente a partir de la Transformada de Fourier de la función de autocorrelación de la señal:

$$S_{xx}(f) = TF\{R_{xx}(\tau)\} = \int_{-\infty}^{\infty} R_{xx}(\tau) e^{-2\pi i f \tau} d\tau$$

Cabe mencionar que el valor de potencia de la componente continua (DC) de la señal es el valor de S_{xx} para $f=0$, es decir, $S_{xx}(0)$. En realidad, el valor de potencia de cualquier frecuencia de la señal que se desee conocer se podrá obtener sustituyendo dicha frecuencia en la integral expuesta.

$$S_{xx}(0) = TF\{R_{xx}(\tau)\} = \int_{-\infty}^{\infty} R_{xx}(\tau) d\tau$$

La potencia total de la señal, como se puede deducir fácilmente, será el resultado de sumar la potencia con la que cada una de las frecuencias presentes en la señal contribuyen a ella. Considerando que la DEP es una función continua, es decir, se trata de una función que varía de forma continua a lo largo del eje X, el cual representa la frecuencia, la potencia total de la señal será la integral en todo el espectro de la DEP, es decir, desde menos infinito a infinito.

$$P = \int_{-\infty}^{\infty} S_{xx}(f) df$$

Las señales de audio que se manejarán estarán limitadas en banda, es decir, su DEP no tendrá valor desde menos infinito a infinito, si no que estará acotada entre dos valores que serán 0 y el doble del ancho de banda de la señal, debido al teorema de Nyquist, por lo que además del gráfico de DEP, será posible calcular la potencia total de señal (de ruido en su caso) que tiene.

Conviene recordar que para caracterizar un proceso estocástico de forma totalmente precisa, habría que disponer de un número de infinito de muestras, al no ser esto posible, lo que en realidad se manejará son estimaciones. Por tanto, siempre que hablemos de ruido blanco, procedente de un generador aleatorio o como idealización del ruido de cuantificación, en realidad, siempre será una aproximación. Se puede añadir, también, que esta cuestión está relacionada con la naturaleza misma del ruido, ya que es fácil comprender que es imposible construir un generador de ruido blanco totalmente perfecto, puesto que para demostrarlo, habría que tener una serie infinita de los números generados, teóricamente, de forma aleatoria.

La estimación de la DEP, teniendo presente la paradoja comentada, en el caso de que se desee emplear de manera práctica para analizar las señales que se manejarán, se puede calcular mediante diversas vías. De forma teórica, se han establecido dos modos básicos: no paramétricos y paramétricos.

Paramétricos

Se supone un determinado modelo de proceso estocástico y se calculan sus parámetros mediante técnicas de predicción lineal.

No Paramétricos

Se trata de obtener el periodograma de un tramo de señal, es decir, calcular la Transformada de Fourier de un fragmento de señal para conseguir una representación de su espectro. Esto se relaciona con el enventanado de señal, que es la técnica que estudia la extracción de tramos de una señal para calcular su

espectro. En la realidad, se manejarán señales digitales, que son señales discretas, por lo que la herramienta a emplear es la DFT (Transformada de Fourier Discreta).

En la implementación del proyecto, para visualizar la densidad espectral de potencia de las señales de interés, que serán, principalmente, casos de ruido de cuantificación presente en diversos puntos de los sistemas, lo que se empleará será un modelo no paramétrico, siguiendo el esquema de aplicar una ventana de la señal, situada en un punto arbitrario (donde le interese al usuario), y sobre la cual, se calculará la DFT para obtener su Densidad Espectral de Potencia o lo que es lo mismo, el periodograma. El objetivo será observar como evoluciona la DEP de la señal y del ruido en diferentes instantes temporales.

Por tanto, como tal, en principio, no se calculará la Densidad Espectral de Potencia de toda una señal, pues no tiene sentido, si no que se calculará una serie periodogramas sobre tramos temporales de la señal de cierta longitud, para ver como evoluciona el ruido de cuantificación en el tiempo. Sin embargo, a pesar de lo comentado, también será útil realizar un promedio de todos los periodogramas de la señal, para considerar, en media, como se reduce el ruido de cuantificación, mediante el método de Welch. El objetivo final será contrastar la evolución estadística del ruido para diferentes periodogramas situados en distintos instantes temporales y el promedio de todos. Cuanto más se aproximen a la hipótesis de ruido blanco el promedio y cualquiera de los periodogramas, mas predecible será el ruido, lo que es una cualidad deseable para la audición.

Este proceso, es preciso señalar, conlleva un dilema relacionado con la relación entre tiempo y frecuencia que se tratará con mas detalle en el apartado de enventanado de señal. No obstante, se puede explicar brevemente que al tomar un fragmento de la señal, el tamaño de la ventana es un parámetro que se elige de manera arbitraria. A mayor tamaño, mayor resolución frecuencial pero menor resolución temporal. Su valor, por tanto, dependerá las características propias de la señal que se pretende observar y como evolucionan en tiempo y frecuencia. Ésta es una cuestión estudiada en profundidad en el ámbito del tratamiento digital de voz, pues interesa determinar cual es el tamaño óptimo de una ventana para estudiar la evolución en el tiempo de las características espectrales de una señal de voz. Dicho estudio se puede tomar como referencia para implementar las

aplicaciones del proyecto, de hecho, muchas de las señales que se utilizarán como base, serán señales de voz [25].

La fórmula empleada para calcular la Densidad Espectral de Potencia a partir de la DFT es la siguiente, siendo x la señal:

$$S(e^{j\omega}) = \frac{1}{N} \left| \sum_{n=1}^N x_n e^{-j\omega n} \right|^2$$

Esto es, el valor absoluto de la Transformada de Fourier elevado al cuadrado y dividido por el número de muestras.

Empleando una notación simplificada:

$$DEP = \frac{1}{N} |X[k]|^2$$

Siendo $X[k]$ la Transformada de Fourier Discreta la secuencia $x[n]$.

Hay que indicar que esta expresión utiliza el factor de ponderación $1/N$, el cual, es válido si se emplea una ventana uniforme para extraer el fragmento. Generalizando para cualquier ventana:

$$DEP = \frac{1}{W'W} |X[k]|^2$$

Donde W es la matriz que representa la ventana y W' su traspuesta. Teniendo en cuenta que W es una matriz columna que contiene los coeficientes que definen la ventana, el cálculo del factor de ponderación es, en la práctica, el sumatorio de los cuadrados de sus coeficientes

$$\text{factor de ponderación} = \frac{1}{\sum_{n=0}^N W(n)^2}$$

El resultado, finalmente, incluyendo todos los elementos:

$$DEP = \frac{1}{\sum_{n=0}^N w[n]^2} |X[k]|^2 \text{ y } X[k] = DFT(x[n]w[n])$$

donde $X[k]$ es la DFT de la señal $x[n]$ ponderada por la ventana $w[n]$

2.1.1.4 La Transformada Rápida de Fourier (FFT)

La densidad espectral de potencia se ha definido como una transformación de la Transformada de Fourier. El cómputo fundamental que hay que realizar, por tanto, para obtener dicha función es aplicar, en primer lugar, la Transformada de Fourier sobre la señal de la que se quiere calcular el espectro. Las señales con las que se trabajará son digitales, dicho de un modo mas elemental, un array de muestras (ver teoría de la conversión A/D), con lo cual, en realidad, lo que se tendrá que computar es la Transformada de Fourier Discreta de una secuencia [4].

La expresión con la que se obtiene, para una secuencia $x[n]$, es la siguiente:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N}nk}$$

El cómputo de la DFT utilizando esta fórmula requiere realizar N^2 operaciones, lo cual conlleva un coste computacional muy alto que dificultaría enormemente su implementación práctica en una aplicación informática, especialmente, si tienen que tratarse grandes cantidades de datos, sin ir mas lejos, por ejemplo, una señal de audio digitalizada.

Para solventar este problema, existe una variedad de algoritmos capaces de computar la FFT con menos operaciones, en concreto, $N\log(N)$. A esta familia de algoritmos que tienen en común reducir el número de operaciones necesarias para calcular la DFT, se les denomina algoritmos FFT (Fast Fourier Transform).

La investigación sobre los citados algoritmos despierta un gran interés desde las últimas décadas, que se mantiene a día de hoy, pues en combinación con los avances en la computación y la informática, abre un campo con enormes posibilidades que ha dado lugar a la implementación práctica de aplicaciones de tratamiento de señal, de muy diferente ámbito (telecomunicaciones, informática, audiovisual, etc) que utilizan las herramientas que proporciona la base matemática del análisis de Fourier gracias a la simplificación computacional que aportan los algoritmos FFT.

El método mas ampliamente utilizado y con una mayor difusión es el de Cooley-Tukey, propuesto en los años sesenta y que fue ideado a partir de trabajos anteriores de Gauss.

La idea esencial que toma este algoritmo como base para agilizar el cálculo de la transformada de Fourier, es dividir una secuencia de N términos que se quiere transformar, en otras dos, de N/2 términos, de modo que se toman las muestras pares de la secuencia y se calcula su DFT con la fórmula tradicional, y análogamente, se hace lo mismo para las muestras impares, para luego, combinar ambos resultados y encontrar la DFT de la secuencia original de N términos. Más aún, este método se aplica de forma recursiva descomponiendo sucesivamente en secuencias cada vez con un menor número de muestras, reduciendo de esta manera el número de operaciones, ya que el conjunto consistente en calcular la DFT de una serie de secuencias pequeñas y su combinación posterior, conlleva mucho menos coste computacional que tomar directamente la secuencia de N muestras y aplicar la fórmula de la DFT.

De modo matemático, se expresa el cómputo de la DFT de una secuencia de N muestras, mediante la combinación de la DFT de dos secuencias de N/2 muestras, de esta manera:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N}nk} = \sum_{m=0}^{\frac{N}{2}-1} x_{2m} e^{-\frac{2\pi i}{N}(2m)k} + \sum_{m=0}^{\frac{N}{2}-1} x_{2m+1} e^{-\frac{2\pi i}{N}(2m+1)k}$$

donde los términos 2m corresponden a las muestras pares y 2m+1 a las impares.

Este proceso se considera comúnmente una variante del método de Cooley-Tukey, de hecho, la mas popular, denominada "Radix – 2".

La ventaja en la reducción del coste computacional del cálculo de la DFT que aporta este algoritmo se puede observar con claridad al hacer una comparación de las funciones N^2 y $N\log(N)$ que describen dicho coste:

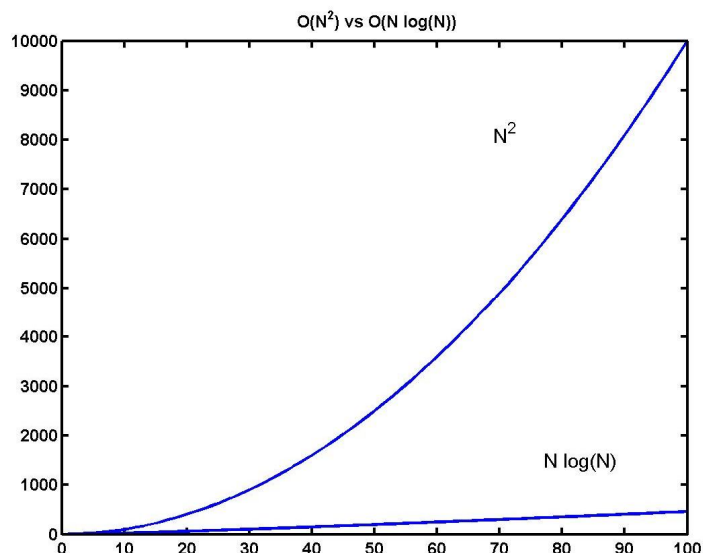


Figura 2-1 Coste computacional FFT

En datos numéricos, se puede comparar la reducción del coste computacional para varios valores de la longitud N de la secuencia sobre la que se aplica:

N	10	100	1000	10^6	10^9
N^2	100	10^4	10^6	10^{12}	10^{18}
$N \log N$	10	200	3000	6×10^6	9×10^9

Para la aplicación concreta para la cual se va a utilizar en el proyecto, se tomarán fragmentos enventanados de señal de unos 20ms, por razones que se expondrán mas adelante, lo cual significa, para una señal de audio digital con frecuencia de muestreo 44100Hz, unas 800 muestras, con lo que se obtiene una reducción del coste computacional en un factor de 275.

La implementación del algoritmo de Cooley-Tukey tiene una gran complejidad, para efectuarlo, se ha tomado como base la codificación en Java del mismo realizada por Robert Sedgewick y Kevin Wayne, de la Universidad de Stanford. Esta implementación asume que el valor N , la longitud de la secuencia,

deberá ser siempre potencia de dos, por lo que la técnica para obtener la FFT de cualquier secuencia, de longitud arbitraria, es rellenar con ceros hasta ajustarse a la potencia de dos más próxima. Este proceso está convenientemente explicado en el capítulo correspondiente a la implementación en Java de las applets y comentado en el propio código.

Además de la codificación del algoritmo DFT por los autores citados, se ha tomado la utilidad creada por ellos mismos para manejar números complejos en Java, pues esta implementación toma como dato de entrada una secuencia de N muestras expresadas en sus términos reales e imaginarios, que forman cada una, un objeto de la clase "Complex", para ello, por tanto, se toman muestras de audio como la parte real y la parte imaginaria se le da valor cero [4] [5].

2.1.1.5 Enventanado de señal

El enventanado de señal es una técnica de procesamiento que consiste en aplicar una función limitada en un intervalo temporal $[0, T]$, multiplicándola sobre la señal de interés que se quiere estudiar. De este modo, se extrae un fragmento de señal para realizar sobre él análisis espectral, dejando a cero todas las muestras que queden fuera del intervalo mencionado.

El objetivo de aplicar una ventana es observar el espectro de un segmento dado de la señal, de un número finito de muestras, de modo que se observen sus características para ese trozo concreto, y aplicando sucesivas ventanas que vayan extrayendo fragmentos consecutivos, apreciar la evolución del espectro en función del tiempo.

El intervalo de señal sobre el que se aplica la ventana es interesante que sea estacionario, que la distribución espectral de potencia se mantenga constante, hecho que se aprecia con claridad al tomar, por ejemplo, el uso del enventanado de señal que se hace en el ámbito del tratamiento digital de voz, disciplina en la que precisamente, esta técnica tiene una mayor relevancia.

El objetivo de enventanar una señal de voz es dividirla en segmentos en los que sus propiedades, es decir, las componentes de frecuencia que la forman, se mantengan constantes. Por ejemplo, una vocal, se compone de formantes, que son una serie de frecuencias que la definen y la distinguen de las demás vocales,

permitiendo al oído identificarla. Durante el intervalo de tiempo en el cual está presente una vocal, por tanto, al inspeccionar su espectro, se verá una silueta invariante, compuesta por los formantes mencionados. Esto es la base del tratamiento digital de voz, identificar los segmentos de señal en los cuales las propiedades se mantienen estacionarias y que pueden caracterizar elementos como una vocal.

Para el proyecto, surge esta cuestión con implicaciones similares. El objetivo es estudiar la acción de las técnicas de digitalización de audio sobre el ruido de cuantificación. Esta acción es diferente según las propiedades de la señal, por ejemplo, no se obtiene el mismo resultado para señales de alto nivel, que oscilan entre los valores máximo y mínimo del fondo de escala, frente a las de bajo nivel, próximas al silencio. Es necesario mencionar además en este punto que una de las premisas del proyecto es utilizar fragmentos de voz como señales de audio para testear las técnicas, pues son una buena base para analizar su efecto. Por tanto, podrán tomarse como referencia las ideas sobre inventariado que se utilizan en el campo del procesado de voz.

En conclusión, el cómputo global de la potencia de ruido de cuantificación presente en toda una señal de voz, de una duración arbitraria, y que puede evolucionar de manera muy variada, dará una idea de en qué medida consigue reducir, en promedio, el error para esa señal la técnica estudiada; pero más aún, es interesante investigar el efecto de la técnica sobre segmentos de audio cortos, con una duración en la que sus propiedades sean estacionarias, para apreciar el efecto de la técnica en función las propiedades específicas de la señal en ese tramo. Para ello, se habilitará un slider con el cual el usuario podrá deslizar la ventana a lo largo de toda la señal de audio, para posicionarla en el instante concreto que le interesa estudiar, observando entonces la densidad espectral de potencia de señal y ruido para el segmento centrado en ese punto. En el fondo, no es más que una versión controlable por el usuario, del tradicional gráfico de espectro que muestran los reproductores de música y otras aplicaciones de audio mientras se escucha un fichero de sonido. Los detalles sobre la implementación práctica de este subsistema del proyecto se detallarán más adelante en el capítulo correspondiente a la codificación en Java de las applets. A continuación, se

presentan las consideraciones teóricas sobre enventanado que se han tenido en cuenta para llevar a cabo el diseño de dicho componente.

Los parámetros fundamentales que determinan el enventanado de una señal son su longitud temporal, es decir, el tamaño del intervalo extraído, sobre el cual se ha discutido anteriormente, y la función que define la ventana.

La acción de enventanar una señal se define como la multiplicación en el tiempo de la señal por la función ventana:

$$s_w(t) = s(t)w(t)$$

por tanto, en el espectro, el efecto será realizar una convolución entre las transformadas de la señal y la ventana:

$$S_w(f) = S(f) * W(f)$$

La elección de la longitud temporal hay que realizarla, como se ha expuesto antes, en función de las propiedades de la señal y su evolución en el tiempo. Como las señales que se utilizarán para el tutorial serán principalmente de voz, se ha tomado como intervalo de segmentación 20 ms, valor muy usual en aplicaciones de tratamiento de voz, en las que se recomienda entre 10 y 30 ms, pues para una duración menor la variación del pitch es excesiva y para uno mayor, no se pueden reflejar con suficiente precisión los cambios en el espectro de la señal.

La elección de la función ventana determinará como influirán los transitorios que se dan en los extremos del segmento de señal enventanada sobre el resultado final, la densidad espectral de potencia.

La ventana más sencilla que se puede aplicar, de forma inmediata, es la uniforme:

$$w(t) = \begin{cases} 1, & 0 < t < T \\ 0, & \text{resto} \end{cases}$$

lo que supone, simplemente, seleccionar las muestras que pertenecen al intervalo temporal del segmento, es decir las comprendidas entre 0 y T.

Hay que remarcar que el resultado final, el espectro que realmente se observará, como se ha detallado anteriormente, es la convolución de los espectros de señal y ventana. Lo que interesa ver, como es obvio, es el espectro de la señal solamente. Por tanto, será lógico buscar una función ventana que modifique lo menos posible el espectro de la señal de interés al producirse la convolución de ambas. Lo deseable, como se puede deducir, sería utilizar la función delta en el espectro. La convolución de cualquier señal sobre una delta es la misma señal, intacta. Sin embargo, esto es irrealizable en la práctica, pues supondría no afectar a la señal en el tiempo, es decir, no realizar el enventanado realmente.

Para abordar este problema, existe una familia de funciones ventana que permiten realizar una aproximación. Actúan seleccionando las muestras del intervalo $[0, T]$, tal y como hace la ventana uniforme, pero además, suavizan en valor de las muestras al comienzo y al final de la misma. De este modo, en el espectro, esto supone convolucionar la transformada de la señal original por el de la ventana, que consistirá en un lóbulo central y una serie de lóbulos secundarios anexos, lo que en realidad es una aproximación a la función delta de Dirac.

Se pueden citar como algunas de las ventanas más conocidas la Hann, la Hamming, Blackman-Harris, Bartlett, etc. Sin entrar en detalles sobre las características particulares de cada una, es necesario mencionar que no existe ninguna ventana que pueda considerarse la más óptima en términos absolutos, todas ellas implican tomar una decisión de compromiso entre la resolución frecuencial y temporal que se desea obtener, lo cual, depende del tamaño de los lóbulos principal y secundarios de la transformada de cada una de ellas. La ventana Hann es la más común, ya que ofrece unas prestaciones aceptables y un buen compromiso y además, tiene una implementación sencilla [4] [7].

La expresión matemática de la ventana Hann es la siguiente:

$$w(n) = 0.5 \left(1 - \cos \left(\frac{2\pi n}{N-1} \right) \right)$$

donde N es el número de muestras que forman el segmento enventanado

La representación en tiempo y frecuencia, la siguiente:

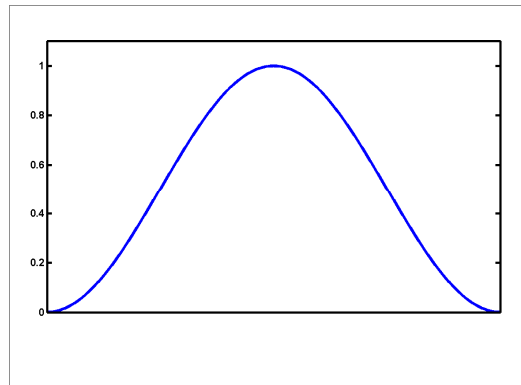


Figura 2-2 Ventana Hann - Tiempo

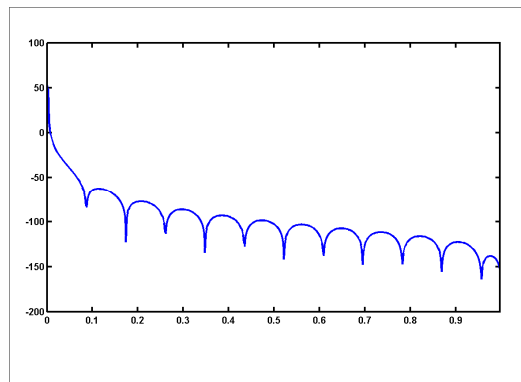


Figura 2-3 Ventana Hann - Frecuencia

2.1.1.6 Cálculo de la DEP promedio. El método de Welch

El método de Welch es una técnica empleada para estimar la densidad espectral de potencia de una señal.

Como se ha explicado previamente, el proceso elegido para observar el espectro de la señal es la división de ésta en ventanas de 20 milisegundos sobre las cuales se aplicará la transformada de Fourier, para de este modo, apreciar las características locales del espectro en el segmento enventanado.

No obstante, surge también la necesidad de presentar el espectro para toda la señal, para evaluar, en promedio, la acción de las técnicas de digitalización de audio. Hay que recordar, para justificar el uso de este método, la relación comentada anteriormente entre la densidad espectral de potencia y los procesos estocásticos. Para ruido uniforme y blanco, su DEP consistirá, idealmente, en una

función plana, no obstante, sería necesario disponer de infinitas muestras para comprobarlo, en la práctica, solo se puede obtener una aproximación, a mayor número de muestras, más cerca estará del caso ideal. Por tanto, para el espectro del ruido de cuantificación de toda la señal, se observará un resultado mucho más cercano al modelo ideal de ruido uniforme y blanco, en comparación con el apreciado para cada ventana de 20 milisegundos.

Para realizar el cálculo del espectro de toda la señal es interesante recurrir al método de Welch, pues es un algoritmo muy sencillo que ofrece un buen compromiso entre coste computacional y prestaciones de la estimación.

Consiste, esencialmente, en dividir la señal en ventanas consecutivas y calcular para cada una de ellas la densidad espectral de potencia. Con el conjunto de transformadas obtenidas, que se denominan periodogramas, se realiza un promedio y se obtiene el espectro de la señal.

El método de Welch se puede aplicar de dos formas, según se apliquen las ventanas que dan lugar a los periodogramas:

- Ventanas no solapadas
- Ventanas solapadas al 50%

La primera opción es aplicable únicamente si se emplea como ventana la función uniforme. En este caso, las ventanas se aplican sobre segmentos consecutivos, sin solape, por ejemplo: 0 a T, T a 2T, 2T a 3T, etc.

En el segundo caso, las ventanas se aplican solapándose al 50 % unas con otras, por ejemplo: 0 a T, T/2 a 3T/2...etc. Este procedimiento es requerido si se emplean ventanas como la Hann, seleccionada para implementar el tutorial; es así porque las ventanas como la mencionada y otras similares, atenúan el comienzo y el final del segmento de audio que extraen, y por tanto, si se realizase el procedimiento indicado en la primera opción, algunas partes de la señal tendrían mayor peso que otras en el promedio final de la densidad espectral de potencia. De este modo, se equilibra el peso de todas las muestras temporales de la señal de audio, y todas contribuyen por igual al resultado final [7] [25].

2.1.2 Filtrado de señal

2.1.2.1 Concepto

Un filtro es un componente de sistemas de procesamiento que elimina una banda de frecuencias del espectro de la señal. Se utilizan en múltiples aplicaciones con el objetivo de seleccionar la parte del espectro que interesa y desechar aquella que no [1] [3] [7].

En los sistemas construidos para implementar las técnicas de digitalización de audio tratadas en este proyecto, aparecen, en diferentes etapas de los mismos, varios filtros, que deberán ser diseñados para incluirlos en ellos. Se presentará, por tanto, una descripción de la función que tendrán estos filtros en los sistemas, con algunos requerimientos en cuanto a sus prestaciones y las posibilidades de implementación, que llevarán al diseño final que será codificado en las applets. Previamente, es conveniente realizar una breve exposición de la clasificación y características de los filtros en general, que serán tomadas en cuenta a la hora de diseñar, en concreto, los filtros necesarios para los sistemas de las técnicas de digitalización de audio.

Los filtros pueden clasificarse, según su respuesta en frecuencia, es decir, la banda de frecuencias que eliminan:

- Filtros paso bajo
- Filtros paso alto
- Filtros paso banda

Los filtros paso bajo son aquellos que eliminan la banda de frecuencias superior, desde cierta frecuencia de corte, por lo que sólo dejan pasar las bajas frecuencias. Los filtros paso alto, por el contrario, eliminan las bajas frecuencias, comprendidas entre cero y cierta frecuencia de corte. Los filtros paso banda, finalmente, sólo dejan pasar la señal presente entre dos frecuencias definidas, pueden verse como una combinación de varios filtros paso bajo y paso banda.

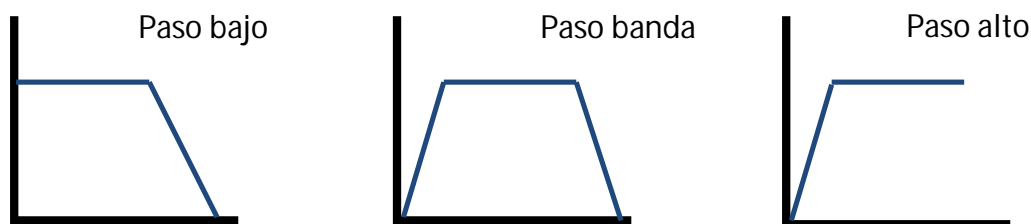


Figura 2-4 Tipos de filtros

Según su implementación:

- Filtros analógicos
- Filtros digitales

El tipo de implementación indica el tipo de señales sobre el cual opera el filtro. Los filtros analógicos se construyen para el procesamiento de señales analógicas, y sus componentes son elementos de circuitos electrónicos analógicos. A su vez, estos filtros, pueden clasificarse en activos y pasivos, según el tipo de elementos que incluyan, como amplificadores operacionales en el primer caso, y condensadores y resistencias en el segundo.

Los filtros digitales se implementan mediante electrónica digital, como DSP's y circuitos integrados, y software. Operan sobre secuencias, es decir, señales digitalizadas.

La caracterización de los filtros se realiza mediante la función de transferencia. Ésta tiene dos componentes fundamentales: la respuesta en magnitud y la respuesta en fase. La primera indica para qué frecuencias y en qué medida para cada una, el filtro aplica atenuación o ganancia. Esta herramienta, por tanto, es útil para clasificar el filtro según los tipos comentados anteriormente (paso bajo, paso alto...etc), pero además, permite conocer con mayor precisión cuál será el efecto para cada zona del espectro o frecuencia concreta, y en base a ello, decidir si el filtro es adecuado para el propósito que se le quiere dar. La segunda indica la distorsión de fase que introduce el filtro para cada frecuencia, en este caso, el objetivo es obtener filtros que provoquen la menor distorsión de fase posible, pues siempre es un efecto indeseado.

La función de transferencia se define, de forma matemática, mediante la transformada Z, en forma de fracción. En ésta, los coeficientes b_k y a_k son los que

determinan el comportamiento del filtro (en magnitud y fase, como se ha descrito antes) y por tanto, su cálculo es la clave para obtener el filtro deseado.

$$H(z) = \frac{\sum_{k=0}^K b_k z^{-k}}{\sum_{k=0}^K a_k z^{-k}}$$

Además, hay que señalar que los valores de b y a que hacen que la función H valga cero se denominan ceros y los valores que hacen que valga infinito, los polos. A partir de ellos se puede construir la ROC, una gráfica bidimensional, de polos y ceros (que tendrán parte real y parte imaginaria) que también sirve de ayuda para deducir el comportamiento que tendrá el filtro, en especial, permite deducir su estabilidad.

Existen una serie de elementos a tener en cuenta para el diseño de un filtro, observables por su función de transferencia. Pueden citarse como más importantes:

Frecuencia de corte

Frecuencia que se toma como punto de separación entre la banda de frecuencias que se deja pasar y la que se elimina. En los filtros paso banda, serán, como es lógico, dos o más.

Pendiente

La separación entre la banda de paso y la banda eliminada no puede producirse de forma brusca, si no que siempre existirá una zona de transición desde la primera, donde la función vale 1, y la segunda, donde vale 0. Esto provoca que las frecuencias de la banda eliminada cercanas a la frecuencia de corte no se filtren totalmente. Por tanto, lo deseable es obtener la mayor pendiente posible, pues así la transición se producirá de forma más brusca y entonces, se filtrarán las frecuencias que se desean.

Rizados en la banda de paso

Si bien la respuesta ideal de un filtro es la unidad para la banda de paso, es decir, no aplicar ganancia ni atenuación en ella, esto no es posible en la práctica, ya

que aparecen bandas de frecuencia donde se aplica cierta ganancia, especialmente, cerca de la frecuencia de corte.

Orden

El orden de un filtro está marcado por el máximo grado que puede tomar la función de transferencia, el retardo de mayor grado. A mayor orden, mayor complejidad de la función de transferencia. Las prestaciones de un filtro aumentan conforme aumenta su orden, aunque no de la misma manera dependiendo de su topología, cuestión que se analiza a continuación.

2.1.2.2 Estructura de los filtros digitales

Los filtros digitales se construyen como un sistema en el que la salida es una combinación lineal de diferentes retardos ponderados por los coeficientes a y b . Tienen dos variantes fundamentales en función de su respuesta al impulso, la cual, viene dada por la recursividad de la estructura que forman.

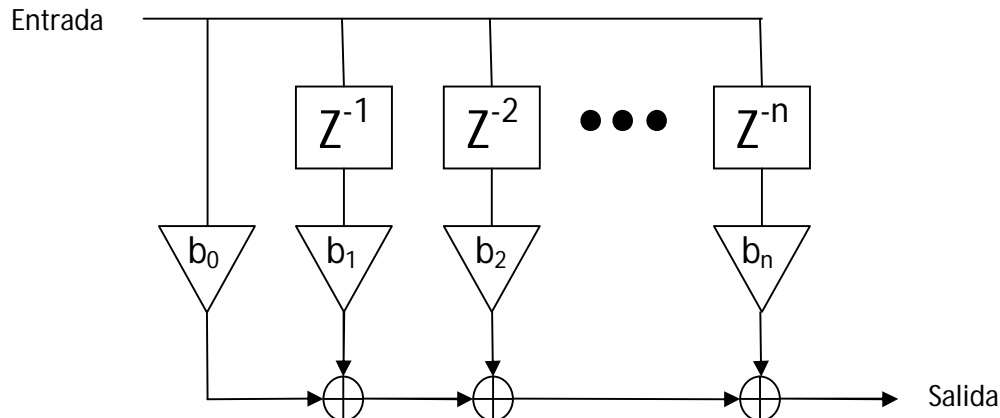


Figura 2-5 Estructura filtro FIR

Los filtros FIR son aquellos que tienen una respuesta al impulso finita, es decir, la señal de salida se hace cero en un tiempo finito, una vez que la señal de entrada ha dejado de introducirse, ya que carecen de realimentación.

Los filtros IIR, por el contrario, tienen una respuesta al impulso infinita, esto quiere decir que la salida será distinta de cero hasta el infinito aún cuando el sistema haya dejado de recibir una señal de entrada. Esto es así porque su estructura es recursiva, parte de la señal de salida vuelve a ser introducida,

sumándola a la de entrada, por lo que el sistema se realimenta, y por tanto, siempre tendrá que existir una salida.

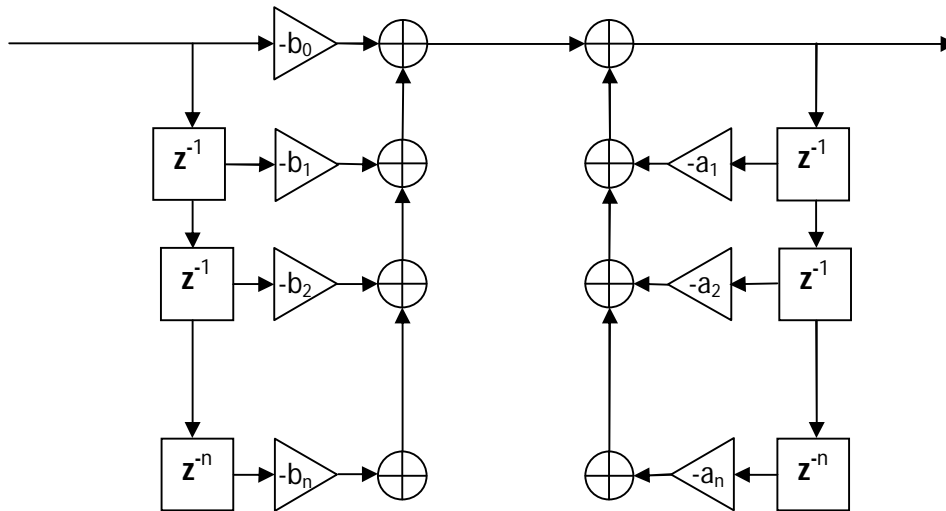


Figura 2-6 Estructura filtro IIR

Estos dos tipos de filtros presentan una serie de ventajas e inconvenientes derivados de su comportamiento, que viene definido por su estructura.

Los filtros FIR son estables intrínsecamente, ya que no hay ninguna posibilidad de realimentación. Ofrecen buenas prestaciones en cuanto a la distorsión de fase, ya que introducen muy poca, es decir, tienen una fase lineal. Sin embargo, su principal desventaja es que requieren un orden elevado en comparación con los IIR, para obtener las mismas prestaciones de la magnitud de la respuesta en frecuencia (pendiente, atenuación de la banda eliminada, etc), lo cual, supone una mayor complejidad en el diseño y fabricación de los componentes electrónicos digitales que los implementan.

Los filtros IIR tienen como principal ventaja ofrecer unas elevadas prestaciones en cuanto a su respuesta en frecuencia para lo cual requieren un orden mucho menor que los FIR. Por el contrario, no garantizan estabilidad, pueden tenerla o no, dependiendo de su distribución de polos y ceros, que se puede inspeccionar mediante la ROC. Para garantizar la estabilidad, es necesario que todos los polos estén contenidos en el círculo unidad. Estos filtros, además,

suelen introducir una considerable distorsión de fase, la cual, puede ser relevante en ciertas aplicaciones [1] [3] [4] [7].

2.1.2.3 Aplicación de los filtros en las aplicaciones y diseño

Los filtros que se han diseñado tienen como finalidad intervenir en los algoritmos de sobremuestreo y Noise – Shaping, eliminando parte del ruido de cuantificación, lo cual provocará una mejoría en la relación señal / ruido de la señal. Las técnicas mencionadas actúan desplazando el ruido a las altas frecuencias, la primera de manera elemental y la segunda mediante un método más sofisticado, pero en definitiva, su objetivo es el mismo, mover la mayor parte de ruido posible a las frecuencias superiores, que gracias al sobremuestreo, quedan fuera del ancho de banda original de la señal, y por tanto, se pueden eliminar sin afectarla. Sin embargo, no hay que olvidar, que por sí mismos, no eliminan el ruido, solo lo mueven. El filtro paso bajo posterior es el elemento que, finalmente, lo eliminará consiguiendo así la reducción esperada de la potencia del ruido de cuantificación. Por tanto, es fácil deducir que las prestaciones del filtro son tan importantes como las de las técnicas tratadas. Aún implementando una función de Noise Shaping de orden elevado y utilizando un factor de sobremuestreo alto, si el filtro paso bajo no funciona correctamente o con la suficiente eficacia, no servirá de nada.

Como se ha dicho, la misión de los filtros es eliminar las frecuencias superiores, derivadas del sobremuestreo. El ancho de banda de la señal se multiplica por el factor de sobremuestreo, que se ha determinado que se seleccione entre los valores 2, 4 y 8. Mediante interpolado, por tanto, una señal con ancho de banda de 20 Khz, por ejemplo, pasará a tener 40, 80 o 160 Khz. Para reconstruir la señal original una vez ha pasado ésta por los algoritmos estudiados, se aplicará un filtro paso bajo que eliminará todo el espectro superior a los 20 Khz de la señal original, que solo puede ser ruido. En ese momento, ya se podrá diezmar la señal. Expresando la frecuencia de forma normalizada, es decir, considerando el ancho de banda de la señal un intervalo entre 0 y 1, es necesario diseñar tres filtros con una frecuencia de corte de $\frac{1}{2}$, $\frac{1}{4}$, y $\frac{1}{8}$.

La primera opción que se puede considerar, la más inmediata, es diseñar un filtro FIR, por la simplicidad de su estructura, ya que no tiene realimentación y

además, por la ventaja que aporta el hecho de que no distorsione la fase, que para una aplicación de audio es especialmente interesante. Sin embargo, para obtener unas elevadas prestaciones, en este tipo de filtros, en cuanto a la pendiente y la atenuación de la banda eliminada, es necesario alcanzar un orden muy elevado, que dificulta su realización.

Los filtros IIR, en cambio, ofrecen una respuesta en frecuencia con unas características muy buenas, utilizando un orden relativamente bajo. Aún así, se considera necesario emplear un orden alto para obtener una atenuación en la banda eliminada lo suficientemente buena para apreciar con claridad el efecto de las técnicas de digitalización de audio.

Para la obtención de los coeficientes que los implementan se ha recurrido a las herramientas que ofrece Matlab para el diseño y estudio de filtros. Hay que recordar que una de las desventajas que, sin embargo, tienen los filtros IIR es su eventual inestabilidad. Para garantizar que son estables, se ha recurrido a la gráfica ROC, en la cual, se ha constatado que todos los polos se sitúan dentro de la circunferencia unidad, además de realizar el suficiente número de tests que han demostrado su buen funcionamiento.

Los coeficientes de los filtros han sido obtenidos mediante la función `iirlpnorm`, que recibe como parámetro el orden deseado del filtro y la descripción de la respuesta en frecuencia buscada, especificando la atenuación o la ganancia deseada para cada zona del espectro, y con ello, devuelve el filtro óptimo. Para los filtros de $\frac{1}{2}$ y $\frac{1}{4}$ se ha determinado un orden nueve, que proporciona entre 60 y 80 dB de atenuación en la banda eliminada, mientras que para el filtro de $\frac{1}{8}$ un orden cinco, que proporciona 40 dB de atenuación, ya que en los superiores se detectaron problemas de estabilidad. Estas prestaciones son más que suficientes para el correcto funcionamiento de las técnicas de digitalización, un rizado en la banda de paso despreciable y una pendiente muy elevada, lo que les convierte en muy adecuados para su inserción en los sistemas del proyecto.

Filtro IIR con frecuencia normalizada $\frac{1}{2}$, diseñado para sobremuestreo factor 2

Respuesta en frecuencia (magnitud)

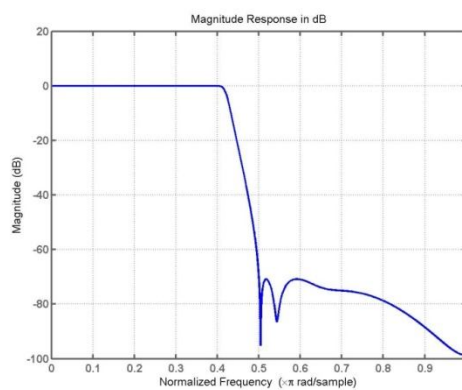


Figura 2-7 Filtro IIR 1/2 (Magnitud)

Respuesta en frecuencia (fase)

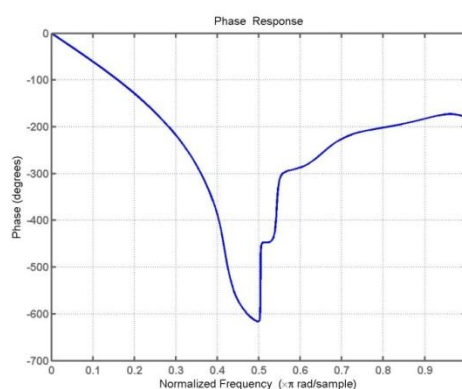


Figura 2-8 Filtro IIR 1/2 (Fase)

Diagrama de polos y ceros

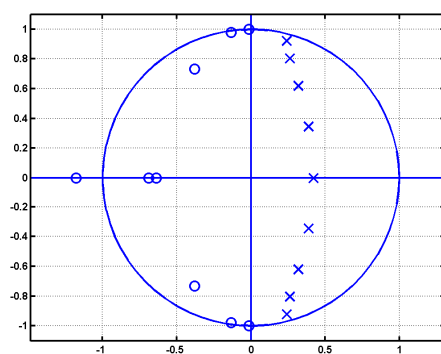


Figura 2-9 Filtro IIR 1/2 (Diagrama de polos y ceros)

Filtro IIR con frecuencia normalizada $\frac{1}{4}$, diseñado para sobremuestreo factor 4

Respuesta en frecuencia (magnitud)

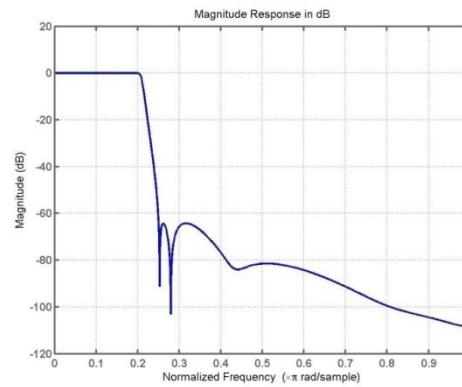


Figura 2-10 Filtro IIR 1/4 (Magnitud)

Respuesta en frecuencia (fase)

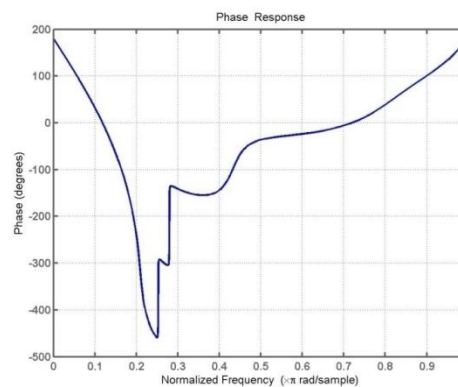


Figura 2-11 Filtro IIR 1/4 (Fase)

Diagrama de polos y ceros

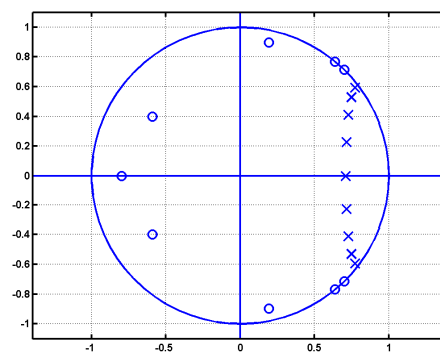


Figura 2-12 Filtro IIR 1/6 (Diagrama de polos y ceros)

Filtro IIR con frecuencia normalizada 1/8, diseñado para sobremuestreo factor 8

Respuesta en frecuencia (magnitud)

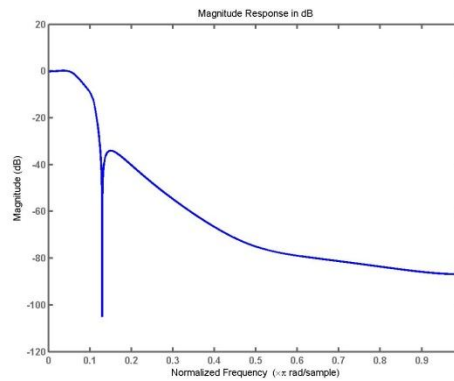


Figura 2-13 Filtro IIR 1/8 (Magnitud)

Respuesta en frecuencia (fase)

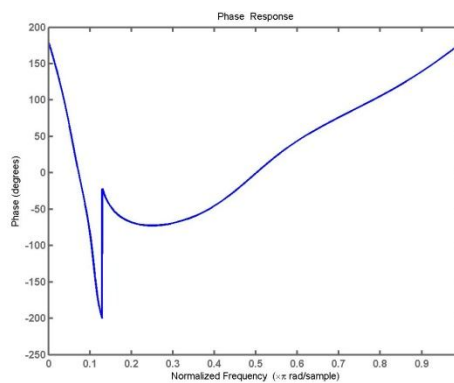


Figura 2-14 Filtro IIR 1/8 (Fase)

Diagrama de polos y ceros

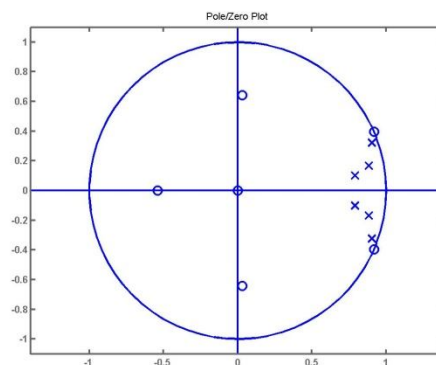


Figura 2-15 Filtro IIR 1/8 (Diagrama de polos y ceros)

2.2 Técnicas de digitalización de audio

2.2.1 Conversión analógico / digital (A / D)

Las señales se dividen en dos tipos fundamentales según su naturaleza.

Las señales analógicas son continuas en el tiempo y pueden tomar cualquier valor de amplitud.

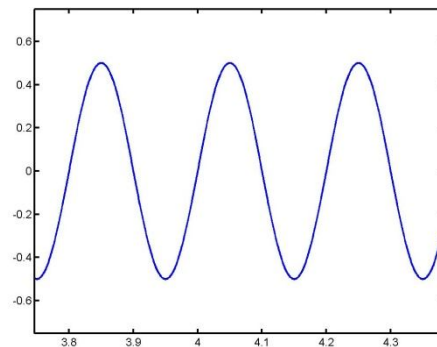


Figura 2-16 Ejemplo señal analógica

Las señales digitales son discretas en el tiempo (solo toman valores a intervalos de tiempo T) y en amplitud (solo pueden tomar niveles dentro de un conjunto finito)

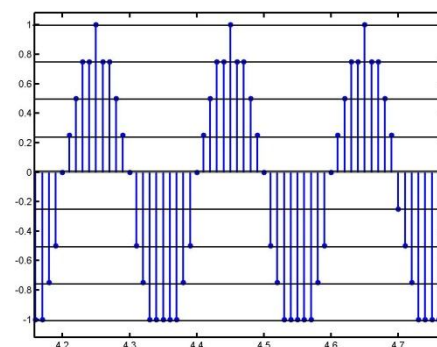


Figura 2-17 Ejemplo señal digital

El objetivo de la conversión A / D es transformar una señal analógica en una señal digital de modo que posteriormente, tras ser transmitida por un canal de comunicación, pueda realizarse la operación inversa en el receptor y se obtenga la misma señal generada en la fuente.

Es decir, en la práctica, se trata de obtener una representación discreta de una señal analógica, es decir, mediante un conjunto de muestras o símbolos definidos, que sólo pueden tomar valores determinados. Se pretende que la señal reconstruida en el receptor guarde la mayor fidelidad posible con la señal original que se digitalizó en la fuente [1] [2] [3] [14] [20].

2.2.1.1 Desarrollo teórico del proceso de conversión A / D

Etapas del proceso de conversión A / D

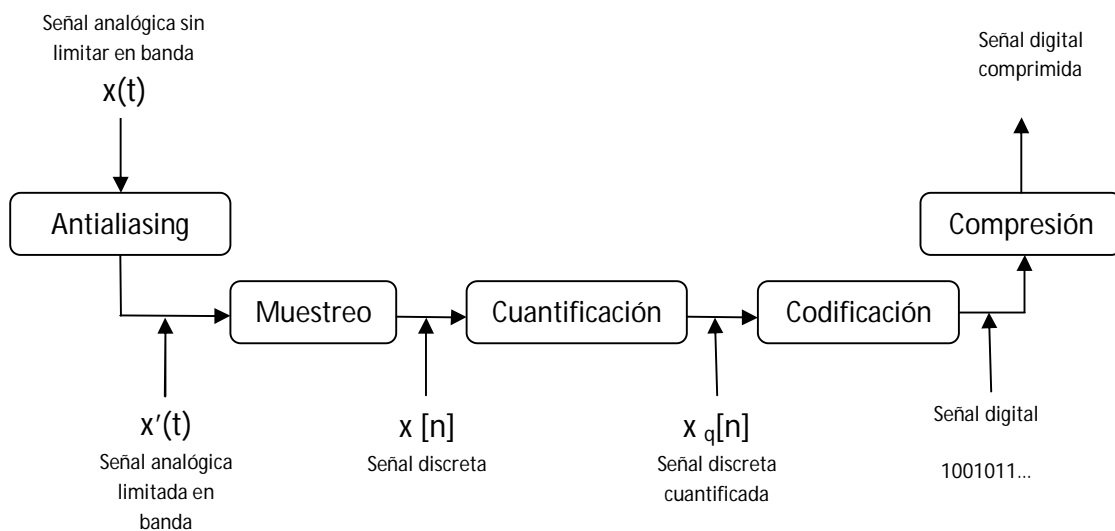


Figura 2-18 Diagrama Conversión A / D

1. Muestreo

Se toman valores de la señal analógica a intervalos de tiempo T . La señal continua queda transformada en una señal discreta, formada por un conjunto finito de muestras, que todavía pueden tomar cualquier valor de amplitud (entre los límites máximo y mínimo de la señal).

Señales analógica y muestreada

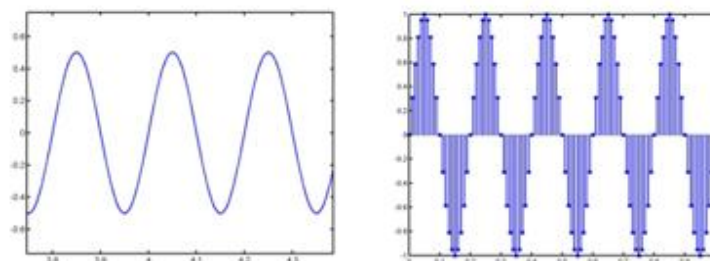


Figura 2-19 Paso de señal analógica a muestreada

El parámetro fundamental de esta etapa es la frecuencia de muestreo.

La frecuencia de muestreo es el valor inverso ($f = 1/T$) del período de muestreo, el espacio de tiempo entre el cual se adquiere cada muestra.

El teorema de Nyquist demuestra que la frecuencia de muestreo debe ser el doble del ancho de banda de la señal original, es decir, el doble de la frecuencia máxima presente en ella, para que posteriormente se pueda reconstruir de forma exacta en el proceso de conversión D / A.

La señal original, por tanto, debe ser limitada en frecuencia por su propia naturaleza, y si esto no es así, debe ser filtrada paso bajo previamente a la etapa de muestreo. Este tipo de filtros se denominan “**filtros antialiasing**”.

La etapa de muestreo puede verse como la convolución de la señal original con un tren de deltas, situadas en múltiplos de la frecuencia de muestreo. El resultado es, por tanto, una serie de réplicas de la señal original centradas en múltiplos de la frecuencia de muestreo.

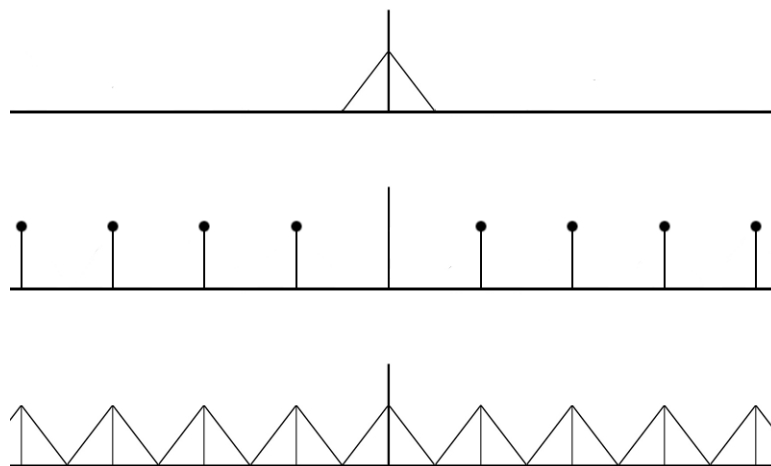


Figura 2-20 Paso de dominio analógico a digital (espectro)

La operación inversa, la conversión D / A, consiste, sencillamente, en el filtrado paso bajo de la señal muestreada para eliminar las réplicas y obtener, de nuevo, una señal analógica.

Si el Teorema de Nyquist no se cumple, las réplicas generadas se solaparán y se mezclarán sus extremos. Este fenómeno se denomina **aliasing**, y da lugar a que la señal resultante de la conversión D / A no sea la señal original, si no que esté

distorsionada. Es por ello que la aplicación de los filtros antialiasing comentados antes es fundamental para que el proceso de muestreo se realice correctamente.

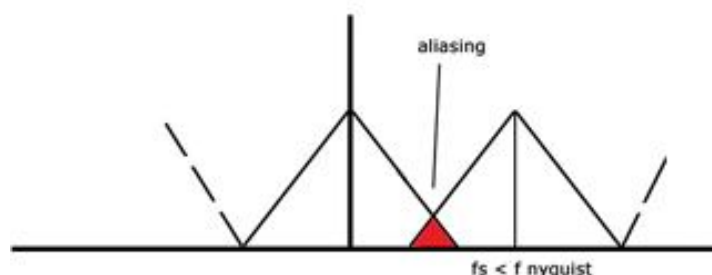


Figura 2-21 Aliasing

La retención de muestras (implementación del muestreo)

La etapa de muestreo se implementa, en los circuitos de conversión A / D, mediante la técnica de **retención de muestras**.

En la práctica, es imposible obtener valores discretos de la señal original a intervalos regulares de tiempo. En realidad, se mantiene el valor de la muestra durante el período T de muestreo, espacio de tiempo durante el cual se toma su valor para almacenarlo y pasarlo a la etapa posterior, la cuantificación.

Esta técnica no tiene consecuencias prácticas desde el punto de vista del estudio teórico de la conversión A / D, pues el resultado, finalmente, es el mismo: un conjunto de muestras de la señal con un valor de amplitud con precisión infinita.

2. Cuantificación

La cuantificación es el proceso en el cual a cada muestra se le asigna un valor de un conjunto finito de niveles. Cada una de las muestras tomadas en la etapa de muestreo todavía pueden tener de amplitud un valor con precisión infinita (es decir, sin límite de decimales). En esta fase se define un esquema de cuantificación, que consiste en un conjunto de niveles, y a cada muestra se le asigna el más cercano.

El parámetro fundamental de esta etapa es el número de niveles de cuantificación (o el número de bits que los definen). Cuanto mayor sea el número

de niveles, menor será el error que se produce al asignar a cada muestra el nivel más cercano.

Esta etapa produce el **ruido de cuantificación**, que consiste en una señal de ruido que se suma a la original y empeora la relación señal / ruido (signal / noise ratio, SNR). El error es la diferencia entre el valor real de las muestras y el del nivel de cuantificación al cual han sido asignadas.

Este concepto se estudiará en detalle en el capítulo correspondiente a la cuantificación, realizándose un análisis cuantitativo de la SNR de las señales cuantificadas en función del número de bits

3. Codificación

La codificación es la etapa en la cual, una vez se ha obtenido un conjunto discreto de muestras cuantificadas, son asignadas a símbolos o valores que las representan, de modo que luego puedan ser enviados por el canal de comunicaciones o guardados en un dispositivo de almacenamiento. Los símbolos que las representan son valores de código binario, aunque existen muchos otros sistemas de codificación, e incluso, los símbolos binarios pueden agruparse o formar otros (constelaciones, en el campo de la Teoría de la Información).

El hecho de utilizar código binario para codificar los valores discretos de amplitud obtenidos implica que los cuantificadores consten siempre de un número de niveles de cuantificación potencia de dos (2^n).

Nivel	Código
N_1	000
N_2	001
N_3	010
N_4	011
N_5	100
N_6	101
N_7	110
N_8	111

4. Compresión

El conjunto de símbolos obtenido (desde un punto de vista elemental, una cadena de ceros y unos), puede ser sometido a procesos de compresión que eliminen la redundancia existente en la información digital y conseguir una reducción del volumen total de la misma. Este es un proceso auxiliar de la conversión A / D, en sentido estricto, no pertenece a la misma, pues el objetivo es exclusivamente obtener una representación discreta de la señal analógica, punto que se alcanza en la etapa 3 (codificación).

No obstante, es necesario mencionar esta acción dentro del estudio de la conversión A / D, ya que en la mayoría de las aplicaciones y sobretodo, en su implementación real, es una cuestión directamente relacionada con la conversión A / D. De hecho, la compresión de la información es un tema de estudio esencial en el tratamiento de señales.

Existen dos tipos fundamentales:

Compresión sin pérdidas

Se aplican algoritmos de compresión que no provocan una pérdida irreversible de información. La información digital que se recupera es exactamente la misma que la que se introduce en sistema de compresión. Las técnicas utilizadas aprovechan la redundancia entre símbolos, patrones repetidos, la diferente probabilidad de aparición de algunos símbolos frente a otros, etc.

Los algoritmos de compresión sin pérdidas, en general, tienen un carácter universal, es decir, se pueden aplicar independientemente del tipo de fuente que genera la información. Una vez que se tiene un conjunto de símbolos codificados, para el compresor, es irrelevante el hecho de provengan de una fuente de audio, de una imagen, de un fichero de texto, etc.

Algoritmos de compresión sin pérdidas típicos son Huffman y LZW, específicos para audio se pueden mencionar algunos como APE, FLEC.

Compresión con pérdidas

Se desprecia parte de la información de la señal original para reducir el volumen ocupado por la misma, por lo que, en el receptor, es imposible reconstruir la señal de forma exacta, se pierde información de manera irreversible.

Este tipo de compresores actúan analizando la señal para clasificar su contenido y eliminar el menos relevante. Según la tasa de compresión que se establezca, actuarán quitando mayor o menor información.

Su principio de actuación se basa en eliminar primero aquella información que no pueda ser percibida, luego, se puede despreciar la información con menor relevancia, según la penalización que se esté dispuesto a pagar a cambio de reducir el volumen de información.

Los algoritmos de compresión con pérdidas se diseñan de manera específica para cada tipo de fuente, según su naturaleza (audio, imagen, video, voz, etc). En el caso del audio, el sistema más difundido es el mp3 (MPEG), que se basa en compresión psicoacústica. Los sistemas de compresión analizan el espectro de la señal de audio, obteniendo su distribución en tonos y bandas, para después, aplicando modelos psicoacústicos, que tienen en cuenta la percepción del oído, eliminan el contenido que no se puede percibir o es el menos relevante.

2.2.1.2 Digitalización de señales: motivación y ventajas

La digitalización es una técnica esencial, clave en el tratamiento de señal en cualquier campo de las telecomunicaciones [23].

El almacenamiento, procesado y reconstrucción de señales digitalizadas presenta numerosas ventajas que han motivado la sustitución progresiva de los sistemas analógicos por los digitales.

Ventajas

Las señales digitales presentan una mayor inmunidad al ruido introducido durante la transmisión por el canal de comunicación. Esta afirmación es relativa ya que, naturalmente, su cumplimiento dependerá de los elementos y circunstancias que entran en juego en el proceso: señal transmitida, parámetros estadísticos del ruido, etc. No obstante, si se puede afirmar que la reconstrucción de señal en el receptor y la protección frente a errores causados por la suma de ruido es mucho más eficaz empleando sistemas digitales que analógicos.

Se puede destacar, por ejemplo, el hecho que el ruido que perturba una señal analógica es acumulativo, es la suma de todo el ruido que ha recogido por su

transmisión por todos los tramos del canal de comunicaciones y los elementos intermedios (amplificadores, multiplexadores, etc).

En el caso de las señales digitales, en cambio, en el paso por dispositivos intermedios la señal puede ser reconstruida y recuperarse de los errores provocados por la acción del ruido.

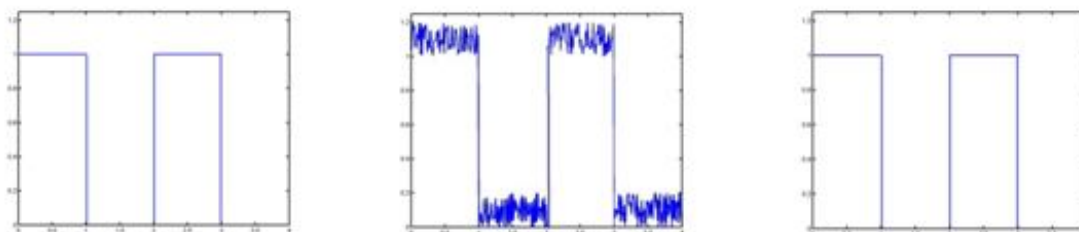


Figura 2-22 Regeneración de señal digital afectada por ruido

El procesamiento de señales digitales ofrece un abanico de posibilidades y opciones mucho más amplio que en el caso analógico. Existe una gran variedad de filtros y efectos que dan resultados muy avanzados, los cuales tienen un diseño más simple y eficaz. Esta ventaja, además, se ve reforzada con la aplicación de la informática en el tratamiento de señal, con los beneficios de versatilidad, sencillez y facilidad que aporta.

El almacenamiento y distribución de contenidos digitales es mucho más seguro, eficaz y sencillo que en el caso analógico. Los sistemas analógicos presentaban numerosos problemas de deterioro y pérdida de información, agravada con el paso del tiempo y el uso. En el caso digital, al tratarse, realmente, de datos, los contenidos se pueden replicar de forma ilimitada, sin deterioro del original, así como persistir en el tiempo sin pérdida de calidad.

La implementación física (hardware) de sistemas electrónicos digitales es mucho más sencilla y menos costosa que los analógicos. Así, se pueden crear dispositivos muchos más pequeños y baratos. En este aspecto, además, hay que tener en cuenta la tendencia de abandonar, en medida de lo posible, la fabricación de aplicaciones en aparatos electrónicos por su implementación a través de software, gracias a la informática, como se ha comentado anteriormente.

Los procesos de multiplexación y modulación de señales digitales presentan, también, numerosas ventajas por su sencillez, coste, etc.

Existen infinidad de ejemplos prácticos donde puede verse la diferencia entre el uso de sistemas analógicos y digitales. En el campo del audio, por ejemplo, en el caso del almacenamiento y distribución de contenidos. Los sistemas antiguos como el disco de vinilo o el cassette presentaban problemas de ruido y deterioro; el sistema de Compact Disc y los archivos digitales, en cambio, permanecen inalterados por tiempo indefinido, se pueden copiar de forma masiva, etc. También en el campo del audio, es destacable la revolución en la creación de sistemas de producción musical, tratamiento y procesado de señal; como dispositivos de efectos sonoros, filtrado, ecualización, mesas de mezclas digitales, etc, que además, pueden fusionarse de forma sencilla con la informática. En el campo de las telecomunicaciones, la digitalización ha dado lugar a sistemas de transmisión de datos, televisión, radio y telefonía (TDT, 3G, satélite, VoIP) con mucha mayor calidad y ventajas notables respecto a los tradicionales analógicos.

Desventajas

El proceso de digitalización conlleva una inevitable pérdida irreversible de información. El muestreo es una etapa que no introduce error si se cumple el teorema de Nyquist, no obstante, siempre será necesario limitar en banda la señal, y eventualmente, se pueden perder altas frecuencias de interés si se debe muestrear a una tasa máxima.

La etapa de cuantificación introduce siempre error en la señal y hace que en el receptor nunca se vaya a poder reconstruir de forma exacta la señal analógica de la fuente.

Según la aplicación y los requerimientos, este factor debe considerarse como importante o no. En el caso del audio, el efecto negativo de la digitalización es irrelevante. Esto es así ya que, en primer lugar, el oído tiene un límite en su ancho de banda que llega a los 20 KHz, y por tanto, puede utilizarse una frecuencia de Nyquist adecuada; en segundo lugar, la distorsión producida por la cuantificación puede evaluarse cuantitativamente para obtener una relación SNR aceptable para el oído, de modo que el nivel de ruido sea imperceptible.

2.2.1.3 Implementación práctica de la conversión A / D

Si bien la realización práctica de la digitalización de audio queda fuera del estudio teórico del proceso, es conveniente mencionar los sistemas básicos, de uso estándar, que llevan a cabo forma efectiva el proceso, para tener una visión completa y cerrada del mismo [1] [14].

El proceso de conversión A / D se lleva a cabo, en la práctica, mediante el sistema PCM (aunque existen otras muchas posibilidades). El almacenamiento informático posterior del audio digitalizado se realiza, utilizando el formato de archivo WAV. El soporte físico más popular ha sido tradicionalmente el CD, si bien, en los últimos tiempos, está tomando gran relevancia la transmisión de audio a través de redes multimedia, utilizando formatos de compresión psicoacústica con pérdidas como el mp3 [1] [2] [3].

PCM (Pulse Code Modulation)

PCM es el sistema de codificación digital de señales más popular, utilizado de forma general en todos los campos (audio, imagen, etc).

Consiste en la modulación de señales analógicas con un tren de pulsos, es decir, es la implementación práctica del proceso de conversión A / D desarrollado teóricamente [14].

Se aplica de la siguiente manera:

1. La señal analógica es muestreada
2. Las muestras son utilizadas para modular un tren de pulsos. Esta etapa puede realizarse de tres maneras diferentes:
 - PAM: la modulación se realiza modificando la amplitud de los pulsos que componen el tren.
 - PWM: la modulación se realiza modificando la anchura de los pulsos que componen el tren.
 - PPM: la modulación se realiza modificando la posición de los pulsos que componen el tren (amplitud y anchura son constantes).

El sistema PCM utiliza cuantificación uniforme y da como resultado una secuencia de información digital codificada en código binario natural.

Es el componente básico de una gran variedad de técnicas de codificación (que realizan conversiones más avanzadas tras obtener la secuencia de código binario natural) y tiene una notable propiedad de robustez frente al ruido. No obstante, hay que tener en cuenta de que se trata de la opción más elemental, la técnica creada inicialmente, pues codifica independientemente cada muestra, y por tanto, da como resultado una elevada tasa de bits.

Otros sistemas son: ADPCM, ITU G.711 a-law, GSM, ITU G.723 ADPCM (Yamaha)...

WAV (Waveform Audio Format)

Se trata de un formato de archivo de audio que almacena en soporte informático (PC) audio digitalizado sin pérdidas. El código binario se almacena mediante la estructura "Little Endian".

Desarrollado por IBM y Microsoft, es una variante del formato AIFF creado para Macintosh.

Los datos de audio pueden provenir de una amplia lista de sistemas de codificación, si bien la fuente más habitual es el PCM explicado antes.

Ofrece una variedad de parámetros de calidad con los que se puede almacenar audio de diferente calidad:

- Estéreo (2 canales) y Mono (1 canal)
- Tasa de muestreo: 44100 Hz (estándar para música), 11025 Hz (voz)
- Bits de cuantificación: 16 bits (estándar para música), 8 bits (audio de baja calidad)
- Compact Disc

El Compact Disc es el soporte físico de almacenamiento de audio de mayor popularidad y difusión, habiéndose convertido en el estándar para uso doméstico a partir de los ochenta

Fue desarrollado en 1979 por Kees Immink, de Philips y Toshitada Doi, de Sony. Consiste en un disco de plástico de 12 cm que es leído por un láser y puede almacenar hasta 700 Mb. Tiene un uso de carácter general, para datos, audio y video.

En el caso del CD-Audio, se emplea el sistema PCM, utilizando una frecuencia de muestreo de 44100 Hz, lo cual ofrece un ancho de banda de 22050 Hz (algo superior al límite del oído) y 16 bits de cuantificación, lo cual ofrece una relación señal/ruido (SNR) de 98 dB teóricos (90 dB reales). Con estos parámetros, se pueden grabar hasta 80 minutos de audio [1].

2.2.2 La cuantificación

2.2.2.1 Introducción

La cuantificación es una etapa del proceso de conversión de señales A / D (analógico a digital).

El objetivo de la conversión A / D es obtener una representación discreta de una señal analógica, es decir, mediante un conjunto de muestras o símbolos definidos, que sólo pueden tomar valores determinados. Se pretende que la señal reconstruida en el receptor guarde la mayor fidelidad posible con la señal original que se digitalizó en la fuente.

Recordando la explicación de la conversión A / D, la primera etapa del proceso consiste en transformar la señal del dominio continuo al discreto. Este proceso se conoce como **muestreo**, en el cual, se toman muestras de la señal analógica con un período T. Según el Teorema de Nyquist mencionado, la frecuencia de muestreo ($1/T$) debe ser dos veces el ancho de banda de la señal. La señal, por tanto, debe estar limitada en banda, por lo que una etapa anterior inherente, es el filtrado paso bajo de la señal.

La siguiente etapa es la cuantificación. Las muestras tomadas de la señal analógica, de amplitud continua, es decir, con una precisión infinita, se asignan a una serie de niveles de cuantificación, de modo que sólo pueden tomar valores de amplitud de un conjunto definido. Las muestras se asignan al nivel de cuantificación más cercano, para minimizar el error.

La cuantificación supone una pérdida irreversible de información, ya que las muestras asignadas a un mismo nivel de cuantificación pierden la información que las distinguía, y por tanto, se tiene una representación más pobre de la señal.

La precisión de la cuantificación se valora con el número de bits empleado, es decir, la longitud de la "palabra digital" utilizada para codificar cada muestra. Cuanto mayor sea el número de bits, menor será el error.

La cuantificación y la pérdida de precisión que supone se modelan con el llamado **error de cuantificación**.

El error de cuantificación supone entender el proceso como un ruido que se añade a la señal original y que reduce su valor de SNR, es decir, empeora la representación de la señal y por tanto, su escucha, al perderse información.

La relación entre la SNR obtenida y el número de bits empleado es necesaria para establecer el compromiso entre la fidelidad de la señal digitalizada y el volumen de información generado para codificar dicha señal [1] [3] [14] [20].

Esquema de la función de cuantificación

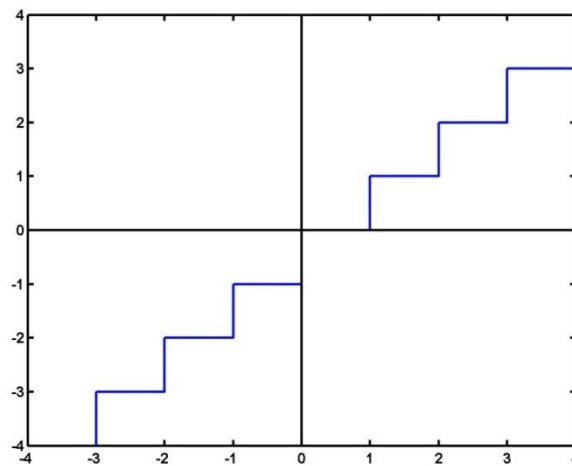


Figura 2-23 Esquema de cuantificación

Parámetros:

- n : número de bits empleado en la cuantificación (longitud en bits de cada palabra con la que se cuantifica una muestra).
- Fondo de escala (FE): espacio entre el valor más alto de amplitud y el más bajo que puede tomar la señal (en el ejemplo, -4 / ; habitualmente, -1/1).
- Paso de cuantificación (ΔQ): tamaño de la franja o espacio en el cual todas las muestras son asignadas a un mismo nivel.
- NL: número de niveles o pasos de cuantificación. Al codificarse en binario, será siempre un valor potencia de dos.

$$\Delta Q = \frac{2X_{max}}{NL} = \frac{FE}{2^n}$$

El paso de cuantificación debe de ser lo más pequeño posible, es decir, el número de bits de cuantificación debe ser el mayor posible.

El ruido de cuantificación (o ruido granular) se da cuando la señal de entrada cae dentro del fondo de escala.

El ruido de sobrecarga se da si la señal de entrada queda fuera de los límites del fondo de escala.

2.2.2.2 Tipos de cuantificación

Según la estructura del cuantificador, se tienen varios tipos de sistemas de cuantificación

Cuantificación uniforme

El tamaño del escalón de cuantificación es el mismo para todos los niveles. Existen dos disposiciones, según exista un nivel de cuantificación situado en cero o no:

- Midrise: no existe nivel situado en cero.

Suele emplearse cuando el número de niveles de cuantificación es par.

- Midthread: existe un nivel situado en cero

Suele emplearse cuando el número de niveles de cuantificación es impar.

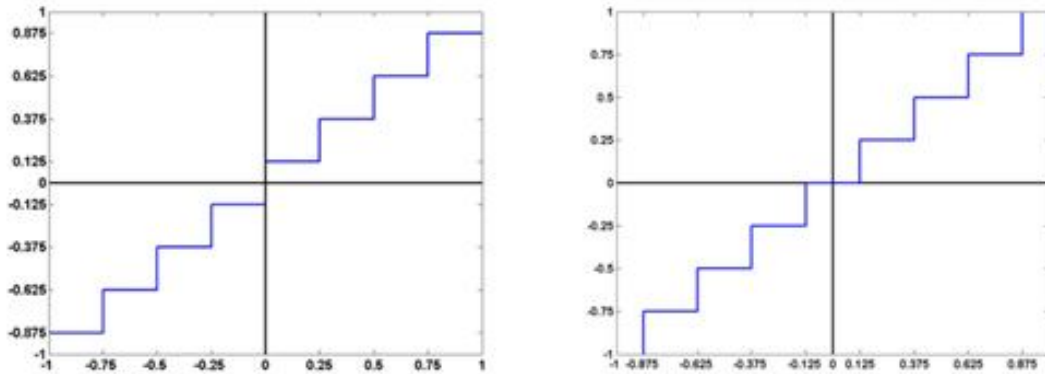


Figura 2-24 Tipos de cuantificación uniforme

Midrise

Mithread

Por ello, el esquema más utilizado es el *midrise*, ya que el número de niveles de cuantificación habitualmente es potencia de 2 (2^n) y por tanto, par.

Cuantificación no uniforme

El tamaño del escalón de cuantificación no es igual para todos los niveles. Este esquema de cuantificación se puede utilizar cuando la señal de entrada tiene una distribución estadística para la cual sea más óptimo emplear un diseño específico, con el que se obtengan valores de SNR mayores que con el empleo de la cuantificación uniforme.

El diseño de un esquema de cuantificación no uniforme es una tarea compleja. En primer lugar, hay conocer la estadística de la señal a cuantificar (su función densidad de probabilidad, y a partir de ésta, los momentos estadísticos). Si esto no se puede determinar, no se puede crear el cuantificador no uniforme. En segundo lugar, habrá que aplicar un algoritmo que calcule la disposición de los niveles, como puede ser, el método Lloyd – Max.

Se puede tomar un ejemplo práctico para comprender intuitivamente, sin entrar en análisis matemático, la aplicación de un esquema no uniforme.

Si la función densidad de probabilidad de la señal de entrada es una laplaciana, los valores que aparecerán con mayor frecuencia serán pequeños (la laplaciana es una función simétrica unimodal de media y moda cero). Por tanto, lo más óptimo será cuantificar con mayor precisión las muestras de magnitud pequeña en contraste con las de magnitud grande, que aparecerán con menor probabilidad. El resultado es un cuantificador con escalones que, desde cero hasta el nivel máximo, tiene una magnitud que se hace mayor logarítmicamente.

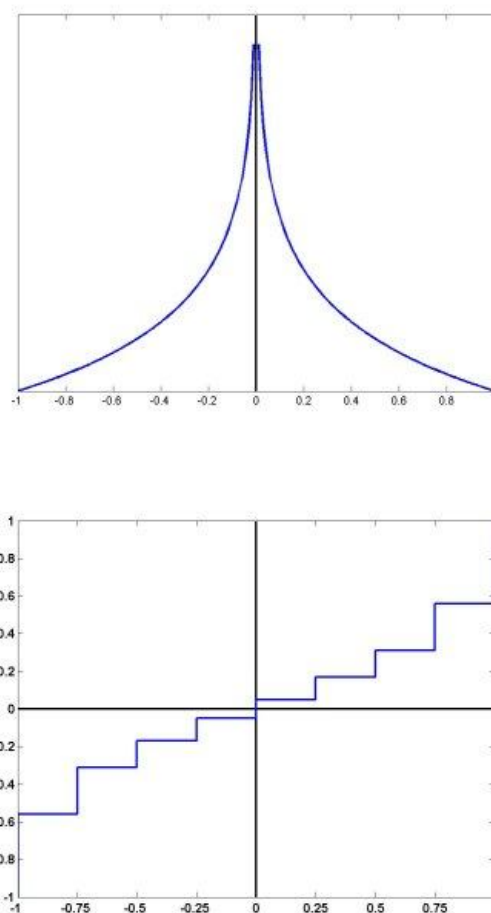
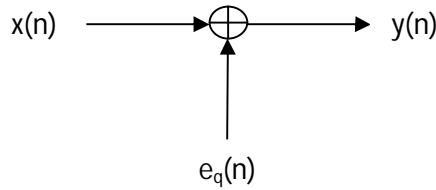


Figura 2-25 Cuantificador logarítmico

Los esquemas de cuantificación no uniforme se utilizan habitualmente en campos como la digitalización de señales de voz, las cuales, pueden ser modeladas estadísticamente. Para casos generales, en los que la señal de entrada no tiene una distribución estadística particular, el esquema uniforme es la única opción.

2.2.2.3 El ruido de cuantificación

El ruido de cuantificación es una señal de error que se añade a la señal original al someterla al proceso de cuantificación. Para cada muestra de la señal original, el error será la resta entre la señal cuantificada y la señal original.



$$e_q(n) = y(n) - x(n),$$

El error de cuantificación estará siempre comprendido entre los valores del escalón de cuantificación.

$$-\frac{\Delta Q}{2} < e_q < \frac{\Delta Q}{2}$$

Se modela estadísticamente como un ruido con fdp uniforme y blanco. Este modelo es sólo una aproximación, suponiendo que la señal de entrada y el ruido de cuantificación generado son independientes, en realidad, puede producirse distorsión si existe correlación entre ambas. Esto se da si existen frecuencias en la señal de entrada múltiplos de la frecuencia de muestreo.

Todo el estudio del proceso de cuantificación está, por tanto, condicionado al cumplimiento de esta suposición [1] [3] [14] [21].

La relación señal / ruido (SNR) es la relación entre los niveles de potencia de la señal de audio original y el ruido de cuantificación que se ha sumado, es decir, el cociente entre los valores eficaces de potencia de dichos elementos al cuadrado.

$$SNR = \frac{S}{E} = \left(\frac{S_{rms}}{E_{rms}} \right)^2$$

Se define una señal de entrada modelo que se supone representativa, para tener un valor de potencia rms concreto en el numerador. Consiste en una onda senoidal, de frecuencia arbitraria, que oscila entre los valores máximo y mínimo

del fondo de escala. Del mismo modo que sucede con el ruido de cuantificación, la validez de las conclusiones a las que se llegue dependerá del cumplimiento de la hipótesis de partida. Se verá más adelante, como en la práctica, para señales de bajo nivel (dado que la señal modelo, justo al revés, ocupa todo el fondo de escala) la validez de los resultados es relativa y hay que tomarlos con precaución.

Los valores máximo y mínimo de la señal de entrada se expresan en función del escalón de cuantificación, para iniciar la demostración:

$$Q2^{n-1}, -Q2^{n-1}$$

El valor de potencia rms de una señal sinusoidal se define como su amplitud máxima dividida por un factor raíz de dos:

$$S_{rms} = \frac{Q2^{n-1}}{\sqrt{2}}$$

El valor E_{rms} se calcula como la varianza del ruido (recordar que se supone uniforme y blanco):

$$E_{rms} = \left[\int_{-\infty}^{\infty} e^2 p(e) de \right]^{\frac{1}{2}}$$

Los límites de integración son $Q/2$ y $-Q/2$ ya que el error de cuantificación sólo puede tomar valores dentro del escalón de cuantificación. El valor de $p(e)$, fdp del ruido, se sustituye por $1/Q$, al ser de tipo uniforme $[(a+b)/2]$ y sale fuera de la integral:

$$\begin{aligned} E_{rms} &= \left[\frac{1}{Q} \int_{-\frac{Q}{2}}^{\frac{Q}{2}} e^2 de \right]^{\frac{1}{2}} = \\ &= \left[\frac{Q^2}{12} \right]^{\frac{1}{2}} = \frac{Q}{\sqrt{12}} \end{aligned}$$

La relación S / E sustituyendo los valores rms calculados es por tanto:

$$SNR = \frac{S}{E} = \left(\frac{S_{rms}}{E_{rms}} \right)^2 = \frac{Q2^{n-1}/\sqrt{2}}{Q/\sqrt{12}} = \frac{3}{2} (2^{2n})$$

Expresando la relación señal / ruido en decibelios, se obtiene:

$$SNR(dB) = 10 \log \left(\frac{3}{2} (2^{2n}) \right) = 6.02n + 1.76$$

De modo aproximado, se puede decir que por cada bits más que se emplee en la cuantificación, mejorará en 6 decibelios la relación señal / ruido.

El resultado obtenido, que es una regla de fácil memorización y ha adquirido una gran popularidad, está sometido, como se ha advertido al comienzo de la demostración, a una serie de objeciones que deben ser tenidas en cuenta para utilizarla de modo correcto y evitar llegar a conclusiones inciertas.

La validez de la ecuación depende de que el ruido de cuantificación cumpla la hipótesis de tener una función densidad de probabilidad uniforme y ser blanco. Expresado de modo matemático, debe cumplir las siguientes condiciones:

1. El error $e_q(n)$ es una señal estacionaria de ruido blanco.
Es decir $e_q(n)$ está incorrelado con $e_q(m)$ para $n \neq m$
2. $e_q(n)$ está incorrelada con $x(n)$
3. $x(n)$ tiene media cero y es estacionaria

En la práctica, se pueden utilizar una serie de reglas intuitivas para comprender con mayor claridad el fenómeno, más allá de condiciones estadísticas de difícil consideración en casos reales.

Si la señal a cuantificar es una onda sinusoidal de nivel muy bajo, tanto, que oscila entre el nivel máximo y mínimo del escalón de cuantificación, el resultado será un pulso cuadrado periódico. El espectro de un pulso cuadrado es una función *sinc*, cuyo espectro es rico en armónicos que distorsionan.

Sin llegar al extremo de considerar un seno de amplitud el escalón de cuantificación, si se observa el espectro de un tono cuantificado, éste consiste en una delta principal centrada en la frecuencia de dicho tono y una serie de réplicas (armónicos del tono) de magnitud menor, que son el error producido por la cuantificación. Estos armónicos distorsionan la señal y empeoran notablemente la audición. En este caso, la hipótesis comentada no se cumple: el error no se

distribuye de manera uniforme en todo el espectro, al contrario, se concentra en unas determinadas frecuencias, y por tanto, la validez de la fórmula que se ha obtenido es discutible.

El mismo fenómeno se produce de manera análoga si considera la situación inversa. Si se tiene una señal sinusoidal de amplitud grande, idealmente, que oscile entre el valor máximo y mínimo del fondo de escala, pero se utiliza un número de bits de cuantificación muy bajo, el resultado será, igualmente, una serie de armónicos que provocan distorsión en vez del ruido con fdp uniforme y blanco que se ha considerado como hipótesis.

El efecto, además, se agrava si el tono tiene una frecuencia submúltiplo de la de muestreo.

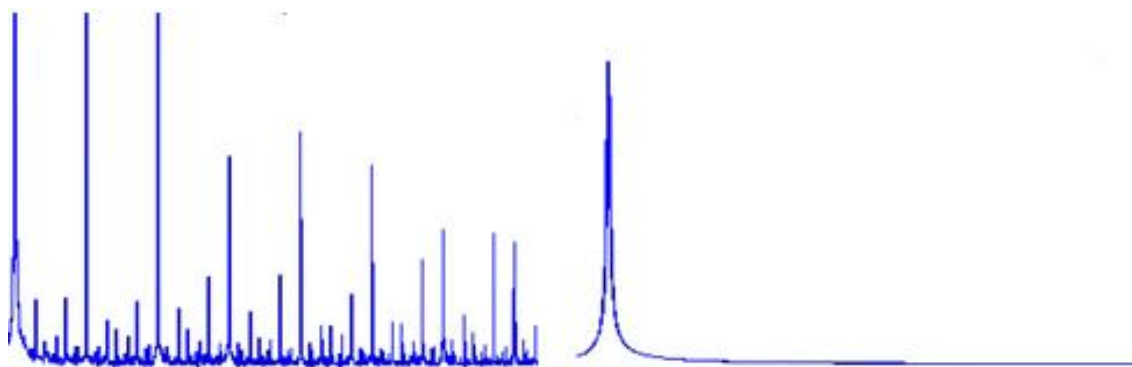


Figura 2-26 Tono cuantificado frente a un tono puro

Tono cuantificado frente a un tono puro. La hipótesis de ruido de cuantificación con fdp uniforme y blanco no se cumple, la potencia del error se concentra en unas determinadas frecuencias (armónicos del tono)

Por tanto, se llega a la conclusión de que para evaluar la validez de la fórmula para cada caso concreto en que se quiera emplear, hay que tener en cuenta la señal y el cuantificador.

Respecto a la señal, cuanto mayor sea su amplitud (su rango dinámico) y que sus componentes en frecuencia no sean submúltiplos de la frecuencia de muestreo, mayor validez tendrá la hipótesis.

Respecto al cuantificador, cuanto mayor sea el número de niveles de cuantificación, mayor validez tendrá la hipótesis.

Aplicación de la fórmula en casos reales

El oído tiene un rango dinámico comprendido entre los 125 y 25 dB SPL, con lo cual, el formato CD cubre todo el rango audible (esta afirmación se hace sin tener en cuenta los procesos a los que es sometida la señal en un equipo de audio, como el *dither*, etc, es una idealización).

En el ámbito profesional, como estudios de grabación de música, se utilizan formatos que emplean un mayor número de bits (24 e incluso 32) para almacenar las pistas que componen una grabación. La relación SNR que proporcionan es muy superior (para 24 bits, 144 dB SPL), exceden notablemente la capacidad del oído, pero su uso es necesario para tener cierta holgura en el proceso de postproducción de una grabación musical.

Una vez el máster está terminado, se procede a recuantificar la señal a 16 bits para codificarlo en CD. En éste paso, se aplican técnicas para preservar la SNR que consiguen valores superiores a los que se tendrían de aplicar el cuantificador directamente.

La distorsión producida por el incumplimiento de la hipótesis es un fenómeno indeseable que empeora de forma drástica la escucha de una señal digitalizada. Su presencia no es permisible pero existen una serie de técnicas que la eliminan, aunque no obstante, a costa de pagar penalización que hay que evaluar.

El *dither* es el sistema fundamental para eliminar la distorsión producida por el ruido de cuantificación. Se estudiará más adelante en profundidad, pero de forma breve, se puede describir como la suma de ruido aleatorio que linealiza el error de cuantificación, eliminando la distorsión, a costa de aumentar la potencia total de ruido de cuantificación.

Fuera del ámbito puramente teórico, es interesante analizar las creencias más extendidas sobre la digitalización de audio y en concreto, la conversión analógica / digital, en el mundo de la música y la grabación de sonido profesional. Existe la idea de que la digitalización de audio provoca un sonido artificial, en contraste con los sistemas analógicos antiguos (como el tocadiscos), que a costa de tener presente un notorio ruido de fondo, reproducían la música de forma

totalmente fiel. Esta afirmación es errónea pues si bien, al cuantificar, aparecen armónicos espurios que efectivamente, distorsionan la señal original, posteriormente se aplican técnicas como el *dither* que lo eliminan. El ruido de fondo de más que supone aplicar el *dither*, si bien, es una penalización, es el mínimo necesario para hacer desaparecer la distorsión, pues se aplica de forma sistemática por medios electrónicos, no se introduce de forma incontrolable por limitaciones técnicas como sucedía con los sistemas analógicos antiguos.

2.2.3 Sobremuestreo

La técnica de sobremuestreo (*oversampling* en inglés) consiste en emplear una frecuencia de muestreo superior a la de Nyquist en el proceso de conversión A / D, la cual, es la mínima necesaria para evitar que se produzca aliasing [1] [3] [14] [15].

Esta técnica puede aplicarse en diferentes etapas del proceso de digitalización con diversos propósitos.

2.2.3.1 Interpolación

El sobremuestreo puede realizarse directamente tomando más muestras de la señal analógica, en el dispositivo conversor A / D, de las que marca la teoría de Nyquist.

Si se tiene una señal muestreada a la frecuencia de Nyquist pero es necesario multiplicar su tasa de muestreo por alguna razón, como aplicar la técnica de oversampling en alguna etapa de la cadena de audio en la que la señal ya está digitalizada, puede aumentarse el número de muestras que la forman mediante **interpolación**.

La **interpolación** consiste en obtener una nueva señal digital intercalando nuevas muestras, calculadas a partir de la propia señal. Se realiza en dos pasos:

1. Relleno de ceros

Se introducen muestras de valor cero entre cada muestra de la señal digital original. Esto resulta en la multiplicación del número de muestras de la señal por un factor. Por ejemplo, si se introduce una nueva muestra entre cada muestra de la señal original, se multiplicará por 2 la frecuencia de muestreo.

2. Filtro de interpolación

Se aplica un filtro paso bajo que suaviza la forma de onda de la señal, su efecto es dar valor a las muestras de valor cero introducidas.

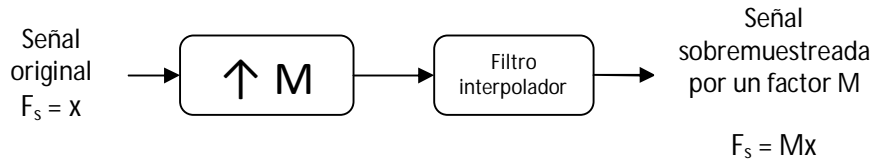


Figura 2-27 Diagrama de la interpolación

2.2.3.2 Efecto del sobremuestreo sobre la SNR

El error de cuantificación y el ancho de banda de una señal están relacionados por una función de área constante. Si el ancho de banda aumenta, el ruido de cuantificación disminuirá de forma proporcional.

Para aumentar el ancho de banda de la señal, se puede aplicar la técnica de sobremuestreo explicada, a través de interpolación o mediante el aumento de la tasa de muestreo en el conversor A / D.

El resultado es que si se duplica la frecuencia de muestreo de la señal, el error de cuantificación disminuirá 3 dB, lo cual es equivalente a emplear 1 bit más de cuantificación.

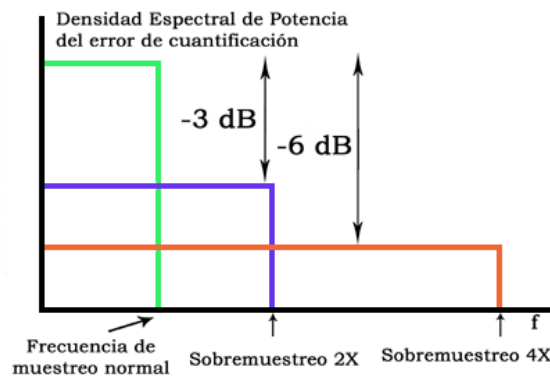


Figura 2-28 Efecto del sobremuestreo

Este fenómeno se produce bajo la condición de que el error de cuantificación se distribuya de igual forma por todo el espectro de la señal, es decir, de que se trate de ruido con densidad de probabilidad uniforme y blanco. Si esta condición no se cumple, el efecto comentado no se producirá.

Por tanto, tal y como sucedía en el estudio de la cuantificación con la ley que relacionaba SNR y bits, la validez del efecto dependerá de la señal y del cuantificador. Cuanto más bajo sea el nivel de la señal de entrada y menor el número de bits de cuantificación, el efecto se producirá en menor medida.

2.2.3.3 Aplicación del sobremuestreo

Aumento de la relación señal / ruido y su aplicación en recuantificación.

El efecto sobre el error de cuantificación puede aprovecharse para aumentar la relación señal / ruido cuando sea necesario, por ejemplo, en casos en los que la señal tenga que ser sometida a una cuantificación concreta y no se quiera perder calidad o en aplicaciones de audio con altos requerimientos.

Esta circunstancia es especialmente útil en procesos de recuantificación, cuando se tiene una señal ya digitalizada. Si la señal tiene que pasar por una etapa en la que el dispositivo (una DSP o más en general, un componente electrónico) sólo puede funcionar con una determinada longitud de palabra digital (bits de cuantificación por muestra), puede aplicarse sobremuestreo para que no se produzca una merma en la calidad de la señal (relación SNR).

Es importante señalar esta ventaja del sobremuestreo. Por un lado, es posible obtener una señal con una SNR superior si se utiliza una tasa de muestreo mayor que la Nyquist, pero alternativamente, también es posible utilizar esta propiedad para mantener la SNR de una señal digitalizada que tenga que ser necesariamente recuantificada (es decir, truncar algunos de los bits menos significativos de cada muestra) para pasar por algún componente de un sistema de audio.

Conversión A / D

El sobremuestreo se puede aplicar en el proceso de conversión A/D para facilitar la implementación del mismo, en concreto, para poder realizar dicho proceso utilizando filtros analógicos más sencillos.

Si se utiliza la frecuencia de muestreo de Nyquist directamente (44.1 Khz, algo más del doble de los 20 Khz que alcanza el oído), es necesario emplear previamente un filtro antialiasing analógico con una pendiente muy abrupta, de

por los menos 80 dB, que pase de ganancia unidad a cero (atenuación total) entre 20 y 22 KHz. Esto supone tener que diseñar un filtro analógico muy complejo.

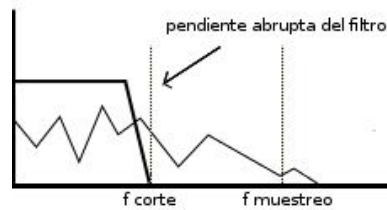


Figura 2-29 Filtro antialiasing de pendiente elevada

Si se muestrea la señal con un tasa superior a la de Nyquist, los requerimientos del filtro se suavizan. Un valor estándar de factor de sobremuestreo que se puede utilizar es 4.

Si se sobremuestra por un factor 4, la primera réplica del espectro quedará centrada en 172,4 KHz ($44.1 \cdot 4$). La señal, por tanto, tendrá que haber sido filtrada para evitar antialiasing cortando a partir de 88.2 KHz. Para ello, el filtro antialiasing tendrá una banda de paso hasta 20 KHz, y a partir de ahí, hasta los 88.2 KHz, una pendiente que puede ser mucho más suave que en el caso simple, por lo que será más sencillo (segundo o tercer orden) y fácil de fabricar.

Posteriormente, en el dominio digital, será necesario aplicar un diezmado para tener como resultado, finalmente, audio muestreado a 44.1 KHz, que es el objetivo. Para ello, en primer lugar, se aplica un filtro digital de orden elevado para eliminar toda la señal que haya quedado entre 20 y 88.2 KHz, pues no interesa ya que queda fuera del rango de audición. Aunque el filtro es complejo y de orden elevado, su implementación tiene mayor viabilidad, y es preferible, por ser digital. En segundo lugar, se diezma la señal por el factor de sobremuestreo empleado (4 en este ejemplo) para obtener un flujo final de 44.1 KHz.

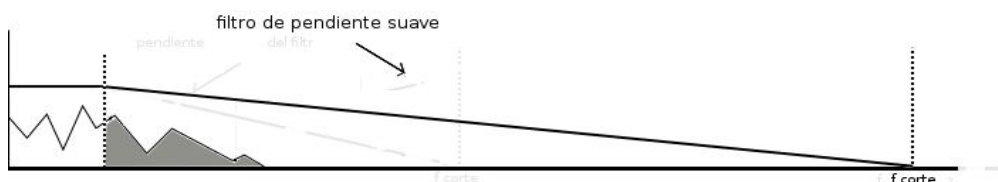


Figura 2-30 Filtro antialiasing de pendiente suave

La salida del sistema, entonces, tanto si se realiza sobremuestreo como si no, es la misma, pero aplicando esta técnica, el filtro analógico antialiasing que hay

que emplear es más sencillo. En cuanto a la toma de muestras a una tasa superior es una cuestión sencilla con la tecnología existente.

Conversión D / A

El proceso de conversión digital / analógico consiste, básicamente, en aplicar un filtro paso bajo analógico, denominado reconstructor, que elimina las réplicas del espectro producidas por el muestreo.

Este filtro presenta la misma problemática que el filtro antialiasing del caso A / D. Es necesario diseñar un filtro con una frecuencia de corte situada en 20 KHz y con una pendiente muy abrupta que finalice en 22 KHz.

Aplicando sobremuestreo se puede mejorar el proceso. Para ello, es necesario multiplicar el número de muestras recurriendo a la **interpolación**, puesto que partimos de una señal digital.

Para ello, se intercalan muestras de valor cero entre las originales y se aplica un filtro paso bajo digital (como se ha explicado en el apartado anterior). Este filtro paso bajo tiene unos requerimientos muy altos, pero al tratarse de un filtro digital es viable.

Una vez se ha aplicado este proceso, la primera réplica de la señal queda centrada en una frecuencia muy superior (factor sobremuestreo * frecuencia Nyquist). Si utilizamos, por ejemplo, un factor de sobremuestreo 4, la primera réplica quedará centrada en $44,1 \cdot 4 = 172.4$ KHz, y dado que el ancho de banda de la señal es de 20 KHz, el espectro comenzará a partir de 152 KHz. Por ello, el filtro reconstructor tendrá que tener una pendiente entre 20KHz y 152 Hz, lo cual supone un diseño mucho más sencillo y de orden más bajo

2.2.4 Dither

El *dither* es una técnica empleada en audio digital para eliminar la distorsión producida por el proceso de cuantificación. Su misión es transformar dicha distorsión en ruido blanco.

El mecanismo esencial del *dither* consiste en sumar a la señal de audio, una señal de ruido, generada de acuerdo a una determinada distribución de probabilidad, escogida adecuadamente.

El uso del *dither* se hace necesario especialmente en procesos de recuantificación, en los cuales, debido a la pérdida de resolución, aparece ruido de cuantificación que no cumple la hipótesis (ruido blanco de distribución uniforme) y por tanto, se produce distorsión [1] [3] [13] [14].

2.2.4.1 El efecto del dither en la audición

La cuantificación (o recuantificación, según el escenario) implica una pérdida irreversible de información, como se ha visto en el capítulo correspondiente.

El análisis matemático del proceso de cuantificación da como resultado una expresión matemática que relaciona la SNR de la señal resultante con el número de bits empleado.

Esta conclusión, sin embargo, no es válida universalmente, para cualquier caso. En la práctica, se comprueba que para señales de bajo nivel en comparación con el fondo de escala, el ruido de cuantificación está correlado con la señal de salida.

La correlación de la señal y el ruido es un efecto muy perjudicial para la audición, que se debe intentar evitar a toda costa. El modelo ideal es tener un ruido de cuantificación con fdp uniforme y blanco, con un nivel que dependerá del número de bits empleado. En términos de estadística, señal y ruido deben ser dos variables independientes.

El *dither* logra eliminar esta correlación a costa de pagar una penalización, la suma de ruido. No obstante, hay que tener presente que para la audición, este inconveniente es aceptable con el objetivo de alcanzar el fin deseado, la

independencia entre señal y ruido de cuantificación. Para el oído, es mejor tener una componente de ruido blanco añadida que percibir distorsión.

2.2.4.2 Estudio experimental del dither

Se tomará una señal senoidal de una determinada frecuencia como ejemplo para ilustrar el mecanismo del *dither*.

Tono de 800 Hz, cuantificado con 16 bits

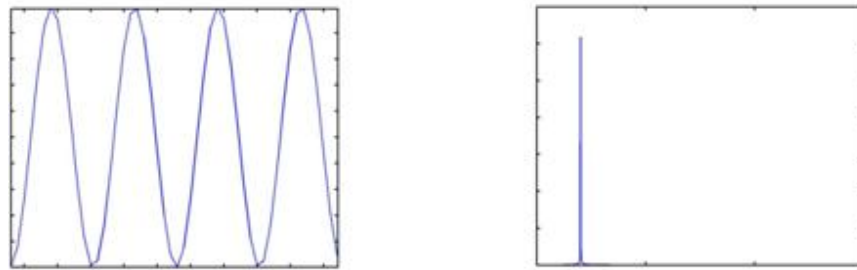


Figura 2-31 Tono de 800 Hz en tiempo y frecuencia

Tono de 800 Hz, cuantificado con 3 bits

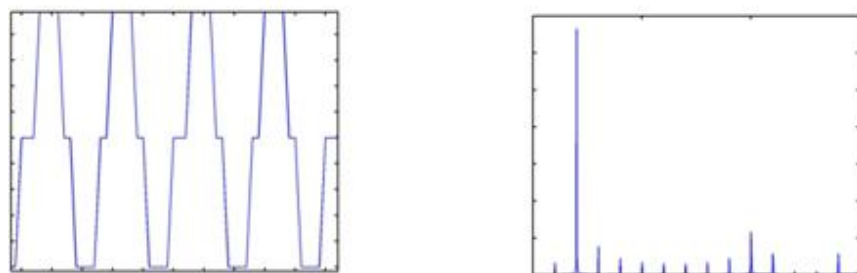


Figura 2-32 Tono de 800 Hz en tiempo y frecuencia

Tono de 800 Hz con dither, cuantificado con 3 bits

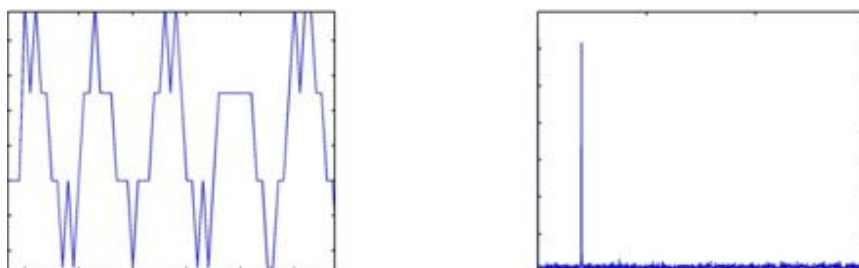


Figura 2-33 Tono de 800 Hz con dither en tiempo y frecuencia

Como se puede ver, el espectro de una onda sinusoidal consiste en una delta centrada en una determinada frecuencia. Al recuantificar, aparecen armónicos, réplicas de la señal original, situadas en múltiplos de la frecuencia de ésta. Con el *dither*, estos armónicos desaparecen, a costa de introducir ruido blanco de cierto nivel.

En tiempo, se observa como los pulsos rectangulares, originados por la cuantificación de señales de bajo nivel, se “aleatorizan”, evitando la aparición de armónicos (la Transformada de Fourier de un pulso cuadrado es la *sinc*)

2.2.4.3 Estudio analítico del *dither*

El *dither* es una técnica conocida desde hace décadas, no es algo reciente. Es un fenómeno físico universal que se puede aplicar en cualquier ámbito: mecánica, electromagnetismo, etc.

Su descubrimiento se sitúa a mediados del siglo XX, en el contexto de la II Guerra Mundial. Se produce de manera accidental, a través de la observación empírica del fenómeno. En el desarrollo de la aviación militar, los técnicos constatan un comportamiento extraño de los elementos que forman la estructuras de los aviones con los que trabajaban, como palancas y engranajes. Su funcionamiento era correcto en vuelo, mientras que en tierra, se producían irregularidades. Llegan a la conclusión de que el comportamiento anómalo se debe a los modos propios que aparecen cuando el avión está en reposo. En vuelo, estos modos propios desaparecen y el comportamiento de las estructuras es ideal. El motivo encontrado era que la fricción con el aire, en el desplazamiento del avión, provoca un temblor, que se podía considerar un ruido aleatorio que impide la aparición de modos propios [1].

En el ámbito del electromagnetismo, y más concretamente, en el audio, el proceso es análogo. Los modos propios (armónicos de las frecuencias que componen una señal) desaparecen cuando se añade una señal aleatoria, ruido blanco.

En una etapa posterior, este fenómeno se aísla y pasa a ser estudiado de un modo analítico. El objetivo es hacer un estudio matemático que explique su funcionamiento.

La demostración analítica del efecto del *dither* se ha realizado por diferentes caminos. Se inicia con los trabajos, no publicados, de Wrigth y Stockham, no obstante, Vanderkooy y Lispshitiz por un lado [16], y Gray y Stockham por otro, a través de diferentes aproximaciones matemáticas, publican teorías que demuestran el funcionamiento del *dither*. Las posibilidades de esta técnica siguen siendo extendidas, con investigaciones como la de Kirac [24].

2.2.4.3.1 Tipos de dither

Existen dos esquemas de aplicación de *dither*:

Dither substractivo

El *dither* es sumado a la señal original. Tras aplicarse la cuantificación, en recepción, se resta el *dither* que se ha sumado anteriormente.

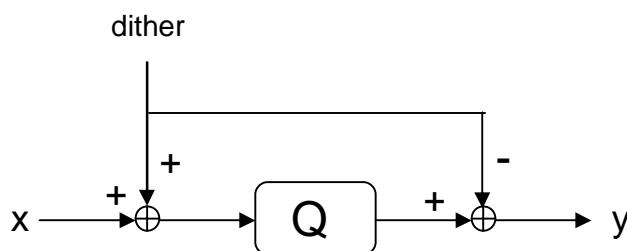


Figura 2-34 Diagrama dither substractivo

Se demuestra analíticamente que con este esquema, se puede lograr la independencia entre las variables señal y ruido de cuantificación, utilizando un *dither* adecuado.

El *dither* substractivo es el utilizado de forma pionera en el estudio de la aplicación de esta técnica. Garantiza la independencia entre señal y ruido de cuantificación.

Presenta un importante inconveniente técnico, que ha provocado que no sea haya convertido en el estándar utilizado. En el escenario habitual de un canal de comunicaciones, implica la obligación de transmitir los valores de *dither* por el canal, para ser luego restados por el receptor, con el consiguiente consumo de ancho de banda, o bien, disponer de un generador de *dither* en el receptor, sincronizado con el de la fuente, lo cual es también problemático.

Dither no-substractivo

Esquema similar al anterior, que no realiza la resta de *dither* a la salida del sistema

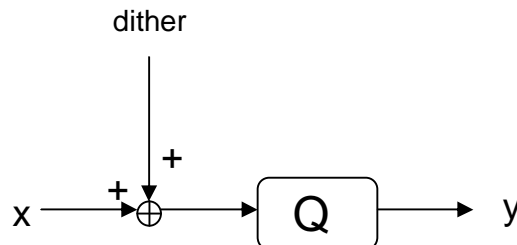


Figura 2-35 Dither diagrama dither no substractivo

Este esquema de aplicación de *dither* no garantiza la independencia entre señal y ruido de cuantificación, pero es más práctico en cuanto a su implementación. Por ello, se hace necesario un estudio analítico que permita lograr resultados satisfactorios, que se aproximen al caso de *dither* substractivo, lo cual, ha sido objeto de investigación posteriormente.

Se ha demostrado, que si bien, no es posible conseguir la independencia entre las variables señal y ruido de cuantificación, es posible conseguirlo con ciertos momentos estadísticos de las mismas, según la función de distribución de probabilidad del *dither* empleado. Por ello, se estudiará el comportamiento estadístico del proceso para lograr un sistema útil en aplicaciones reales

2.2.4.3.2 Análisis del dither

En primer lugar, se establece la condición que se debe cumplir, en el caso de *dither* substractivo, para conseguir la independencia entre señal y ruido de cuantificación

En segundo lugar, se avanza en el estudio, estableciéndose la condición que indica la linealización de ciertos momentos estadísticos según el *dither* utilizado.

Dither substractivo

Se define la función característica P_v como la Transformada de Fourier de la fdp de la señal de *dither*, p_v .

Se demuestra que si P_v cumple la siguiente condición, las variables señal y ruido de cuantificación serán independientes:

$$P_v\left(\frac{k}{Q}\right) = 0, \forall k \in Z_0$$

Q: escalón de cuantificación.

Z_0 : cualquier vector de n componentes enteras.

La función más elemental que cumple la expresión es la distribución uniforme. La Transformada de Fourier de un pulso rectangular es la sinc, función que tiene ceros periódicos en los puntos indicados.

$$TF \text{ de } \left[\prod_q (v) \right] = \text{sinc}(u) = \frac{\sin(\pi Qu)}{\pi Qu}$$

Dither no-substractivo

Se define G_v como la Transformada de Fourier de la fdp de la señal de *dither* p_v , multiplicada por una función sinc

$$G_v(u) = \text{sinc}(u)P_v(u)$$

Se demuestra que si G_v cumple la siguiente expresión, se linealiza el momento estadístico conjunto **m** que relaciona la señal de entrada y el ruido de cuantificación.

$$G^{(m)}_v\left(\frac{k}{Q}\right) = 0, \forall k \in Z_0$$

dónde $G_v^{(m)}$ es la derivada m de la función G_v

2.2.4.3.3 Funciones de densidad de probabilidad de *dither*

Se estudiará una serie de procesos aleatorios para verificar el cumplimiento de las condiciones expuestas anteriormente, con diferentes funciones densidad de probabilidad, para el caso de *dither* no-substractivo [16].

Dither nulo

Se estudia el caso de no añadir *dither* y comprobar que efectivamente, no se linealiza ningún momento estadístico.

La fdp de un *dither* nulo es una delta centrada en cero. La Transformada de Fourier de esta función es la unidad, luego:

$$G_v(u) = \text{sinc}(u)$$

Derivada 1 de $G_v(u)$

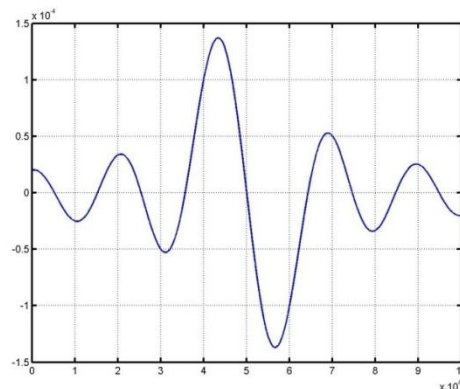


Figura 2-36 Función G_v' para dither nulo

No existen derivadas de G_v que satisfagan la condición (valor cero a intervalos periódicos k/Q , siendo k entero y $Q = 1$), luego no se linealiza ningún momento estadístico y se demuestra entonces, que existe dependencia entre señal de entrada y ruido cuantificación.

Dither uniforme de amplitud 1 LSB

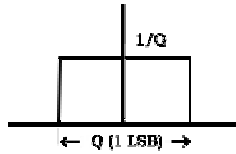


Figura 2-37 Función densidad de probabilidad uniforme

La Transformada de Fourier de la función de densidad de probabilidad uniforme es la función sinc, luego:

$$G_v(u) = \text{sinc}^2(u)$$

Derivada 1 de $G_v(u)$

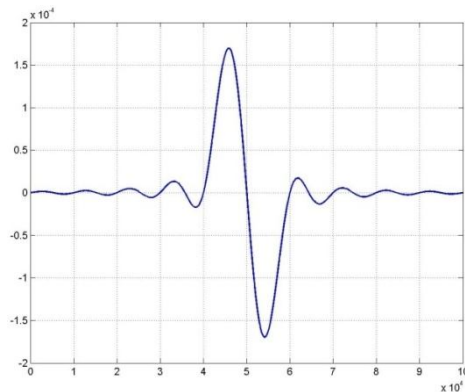


Figura 2-38 Primera derivada de $G_v(u)$ de fdp uniforme

La primera derivada cumple la condición, luego se demuestra que el *dither* uniforme linealiza el momento de primer orden, es decir, la media.

Derivada 2 de $G_v(u)$

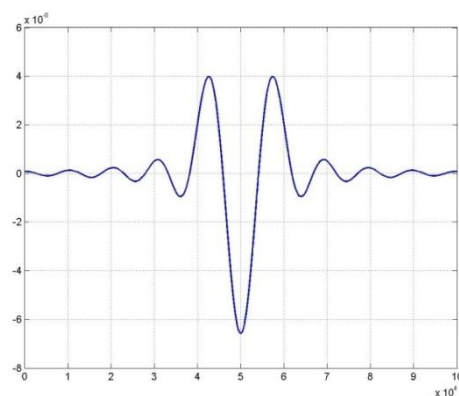
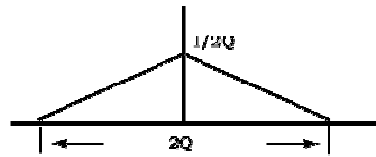


Figura 2-39 Segunda derivada de $G_v(u)$ de fdp uniforme

La segunda derivada no cumple la condición, y por tanto, no se linealiza el momento de segundo orden ni superiores.

Dither triangular de amplitud 2 LSB



La Transformada de Fourier de la función de densidad de probabilidad triangular es la función sinc al cuadrado, luego:

$$G_v(u) = \text{sinc}^3(u)$$

Derivada 1 de $G_v(u)$

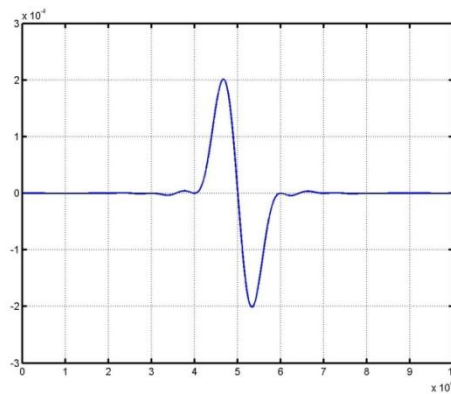


Figura 2-40 Primera derivada de $G_v(u)$ de fdp triangular

La primera derivada cumple la condición, luego se demuestra que el *dither* triangular linealiza el momento de primer orden, es decir, la media.

Derivada 2 de $G_v(u)$

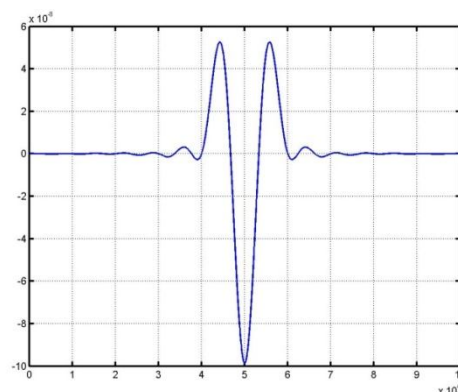


Figura 2-41 Segunda derivada de $G_v(u)$ de fdp triangular

La segunda derivada cumple la condición, y por tanto, se linealiza el momento de segundo orden, es decir, la varianza.

Derivada 3 de $G_v(u)$

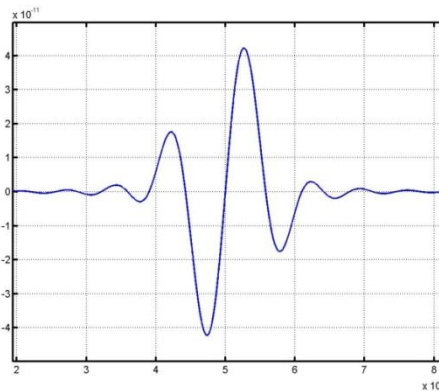


Figura 2-42 Tercera derivada de $G_v(u)$ de fdp triangular

La tercera derivada ya no cumple la condición, y por tanto, el momento de tercer orden ya no se linealiza, ni los superiores.

Otros tipos de *dither*

Dither triangular paso-alto

Se trata de un *dither* de tipo uniforme que es introducido por un filtro paso alto, dando como resultado un *dither* triangular. Este mecanismo aprovecha la característica del oído de tener menor sensibilidad a las frecuencias altas, de modo que el *dither* sea percibido lo menos posible [5].

Gauss

Dither con fdp normal o Gauss. Es sencillo de generar de forma analógica, y de hecho, más que un tipo de *dither*, suele ser un ruido térmico presente en los propios componentes electrónicos de los equipos de audio.

Tiene una potencia muy superior a la del *dither* uniforme y el triangular.

2.2.4.3.4 Potencia total de ruido

El *dither*, como se ha explicado, supone sumar una componente de ruido añadida a la del ruido de cuantificación. En la siguiente tabla se indica la potencia total de ruido en función del escalón de cuantificación [3].

Tipo de <i>dither</i>	Potencia ruido de cuantificación	Potencia de <i>dither</i>	Potencia total
Uniforme	$Q^2/12$	$Q^2/12$	$Q^2/6$
Triangular	$Q^2/12$	$Q^2/6$	$Q^2/4$
Gauss	$Q^2/12$	$Q^2/4$	$Q^2/3$

2.2.4.4 Resumen

El *dither* es una técnica que elimina la distorsión producida por la recuantificación a costa de introducir un componente de ruido añadido.

El objetivo fundamental es conseguir que la señal y el ruido de cuantificación sean independientes. Si existe dependencia, se produce la distorsión comentada, efecto muy desagradable para el oído.

La independencia entre señal y ruido se puede lograr solamente con el *dither* substractivo, pero no es un procedimiento viable. Sin embargo, se ha demostrado práctico utilizar *dither* no substractivo, que si bien no logra la independencia entre señal y ruido, si puede linealizar ciertos momentos estadísticos conjuntos de ellos.

La función densidad de probabilidad que es recomendable utilizar es la triangular, de amplitud 2 LSB. Este *dither* linealiza media y varianza (es decir, hasta el momento de 2º orden). Se ha demostrado en diversos estudios de psicoacústica que el oído es capaz de percibir la correlación entre señal y ruido de cuantificación hasta el momento de 2º orden, por encima es incapaz, luego el *dither* triangular es el más apropiado.

El *dither* es una técnica necesaria en cualquier proceso de recuantificación y es utilizada en múltiples y diversos equipos de audio y de procesamiento de señal. No obstante, para su empleo correcto, hay que tener siempre en cuenta su

funcionamiento. En las situaciones en las que la señal de entrada ya tiene un nivel de ruido gaussiano igual o superior al del *dither* que se sumaría, originado por el ruido térmico de los componentes electrónicos analógicos, por ejemplo, de un estudio de grabación, su aplicación sería innecesaria y contraproducente.

Con el objetivo de reducir al mínimo su impacto en la audición, existen técnicas más avanzadas de *dither*, como el *dither* paso-alto, que aprovecha la baja sensibilidad del oído a las frecuencias altas.

El *dither* suele ir unido a otras técnicas como el noise-shaping, que lo eliminan en parte, tras haber producido el efecto de la "decorrelación".

2.2.5 Noise – Shaping

2.2.5.1 Introducción

El *Noise-Shaping* es una técnica empleada en sistemas digitales de audio para reducir el nivel del ruido de cuantificación generado en procesos de recuantificación.

Consiste en un lazo de realimentación que aplica una función de transferencia que desplaza el ruido de cuantificación a la zona de altas frecuencias, siendo eliminado después mediante un filtro paso bajo (etapa perteneciente ya a la conversión digital-analógica).

El sistema de *Noise-Shaping* es conocido también, en abundante literatura y bibliografía sobre el tema, como "Sigma-Delta Modulation" (modulación sigma-delta). En este caso, la recuantificación se realiza a un bit.

El ruido de cuantificación, según la teoría expuesta en capítulos anteriores, tiene una densidad espectral de potencia uniforme en toda la banda de frecuencias de la señal, tras pasar por la etapa de *Noise Shaping*, parte de la energía se traslada a las altas frecuencias, adoptando una forma de filtro paso alto, con cierta pendiente.

El orden del polinomio que define la función de transferencia marca la pendiente con la cual se desplaza el ruido de cuantificación. A mayor orden, mayor pendiente y por tanto, mayor cantidad es desplazada a las altas frecuencias.

La función de transferencia de un filtro de *Noise-Shaping* tiene dos importantes factores a considerar: la ganancia y la estabilidad. El sistema de *Noise-Shaping*, visto como un filtro, puede no ser de ganancia unidad, por lo que puede no ser estable en determinadas circunstancias, y por tanto, dar como resultado una señal de salida incoherente.

En concreto, si la ganancia del filtro de *Noise-Shaping* es tal que los valores de las muestras superan el fondo de escala (se sobrepasa el rango dinámico del sistema), se produce la inestabilidad comentada y el resultado es negativo.

El orden del polinomio de la función de transferencia y la estabilidad del sistema están directamente relacionados. A mayor orden, mayor ganancia. Es crucial determinar un compromiso entre ambos. Lo deseable es implementar un polinomio de mayor orden posible, pero con la precaución de comprobar su estabilidad.

La técnica de *Noise-Shaping* fue definida por Cutler en 1956. En la actualidad, se persigue obtener nuevos diseños que apliquen una función de transferencia más compleja, de acuerdo a curvas psicoacústicas del oído humano, de forma que se aproveche la diferente sensibilidad a ciertas frecuencias [1] [19].

2.2.5.1.1 Esquema básico de un sistema de Noise-Shaping de orden N

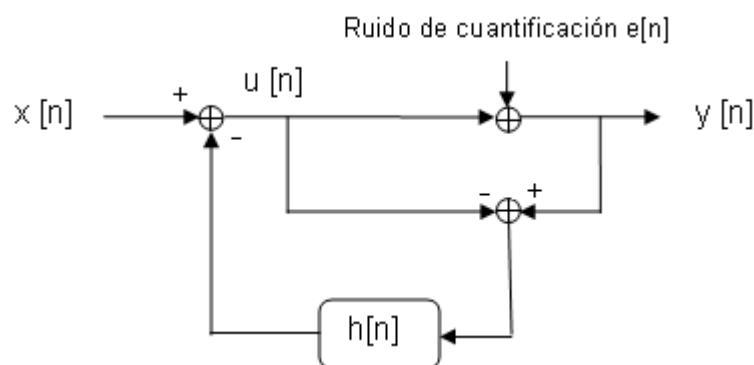


Figura 2-43 Diagrama de Noise Shaping de orden N

Expresión matemática de $h[n]$ en el dominio Z

$$H(z) = 1 - (1 - z^{-1})^N$$

El esquema básico de *Noise-Shaping* da lugar a diferentes versiones del mismo según el orden utilizado [1] [17] [19].

El orden de la función de transferencia determina la forma de la respuesta en frecuencia, es decir, cantidad de ruido desplazado a las altas frecuencias.

Es importante resaltar que el elemento z^{-1} representa un retardo, en un circuito digital, un registro de memoria que guarda el valor de una muestra durante un período. A mayor orden, se guardan muestras en memoria durante mayor tiempo (durante más períodos de reloj del sistema digital).

2.2.5.1.2 Análisis del sistema

Ecuaciones que describen el sistema

$$u(n) = x(n) - h(n) * [y(n) - u(n)]$$

$$y(n) = u(n) + e(n)$$

Transformación al dominio Z:

$$U(z) = X(z) - H(z)[Y(z) - U(z)]$$

$$Y(z) = U(z) + E(z)$$

Se despeja $U(z)$ de la primera ecuación:

$$U(z) = \frac{X(z) - H(z)Y(z)}{1 - H(z)}$$

Se sustituye en la segunda:

$$Y(z) = U(z) + E(z) = \frac{X(z) - H(z)Y(z)}{1 - H(z)} + E(z)$$

Se simplifica:

$$Y(z) = X(z) + [1 - H(z)]E(z)$$

En el dominio de la frecuencia:

$$Y(e^{j\omega}) = X(e^{j\omega}) + [1 - H(e^{j\omega})]E(e^{j\omega})$$

El desarrollo matemático demuestra que el efecto del sistema es actuar sobre el ruido de cuantificación **E**, mientras que la señal de entrada **X** no es afectada.

El factor **H** es el responsable de la modificación del espectro del ruido de cuantificación. Se trata de un filtro IIR de tipo paso alto, por lo que su efecto es desplazar el ruido a las altas frecuencias.

La elección de la función de transferencia **H** es un punto crucial en los sistemas de *Noise-Shaping*. A continuación, se mostrará el desarrollo para varios órdenes de la expresión clásica, aunque no obstante, investigaciones en esta materia se dirigen a encontrar funciones que ponderen la respuesta en frecuencia del oído humano, de tal modo que el ruido se acumule en las bandas donde el oído es menos sensible [1].

2.2.5.1.3 Desarrollo de la expresión $h[n]$

$n = 1$

$$Y(z) = X(z) + [1 - z^{-1}]E(z)$$

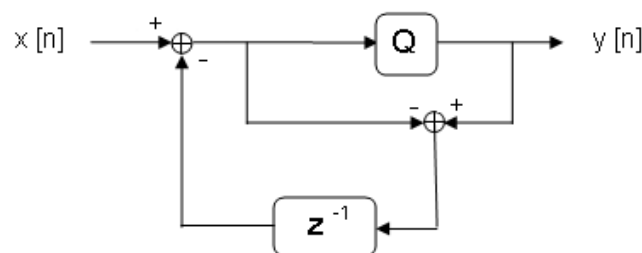


Figura 2-44 Diagrama de Noise Shaping de orden 1

$n = 2$

$$Y(z) = X(z) + [1 - z^{-2} + 2z^{-1}]E(z)$$

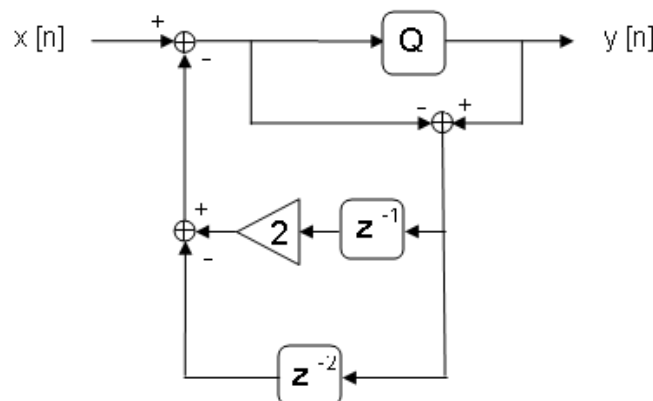


Figura 2-45 Diagrama de Noise Shaping de orden 2

$n = 3$

$$Y(z) = X(z) + [1 + z^{-3} + -3z^{-2} + 3z^{-1}]$$

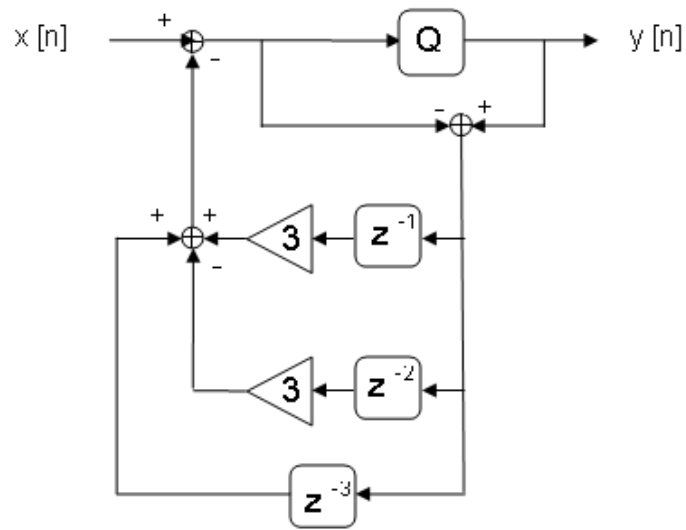


Figura 2-46 Diagrama de Noise - Shaping de orden 3

2.2.5.1.4 Análisis de $H(z)$

Se presentan las gráficas de respuesta en frecuencia, respuesta en fase y diagrama de polos y ceros que caracterizan las funciones expuestas.

$n = 1$

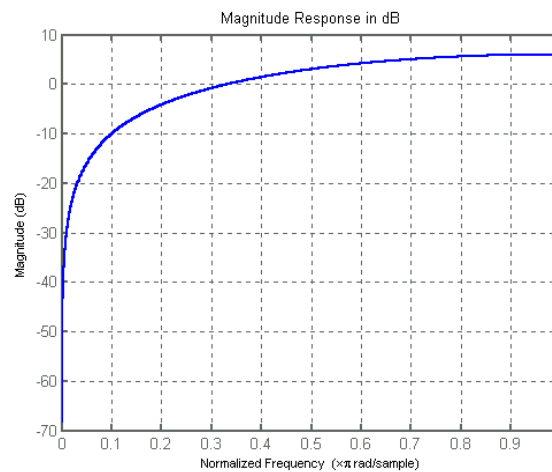


Figura 2-47 Respuesta en frecuencia (Magnitud) de un filtro de Noise - Shaping de orden 1

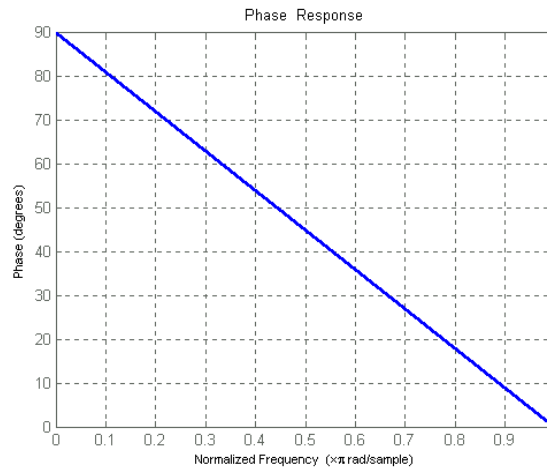


Figura 2-48 Respuesta en frecuencia (Fase) de un filtro de Noise - Shaping de orden 1

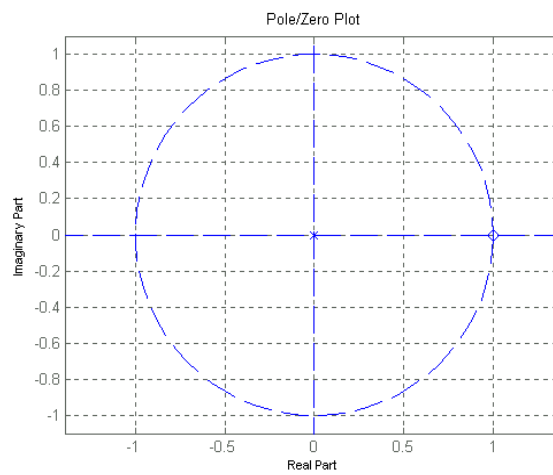


Figura 2-49 Diagrama de polos y ceros de un filtro de Noise-Shaping de orden 1

$n = 2$

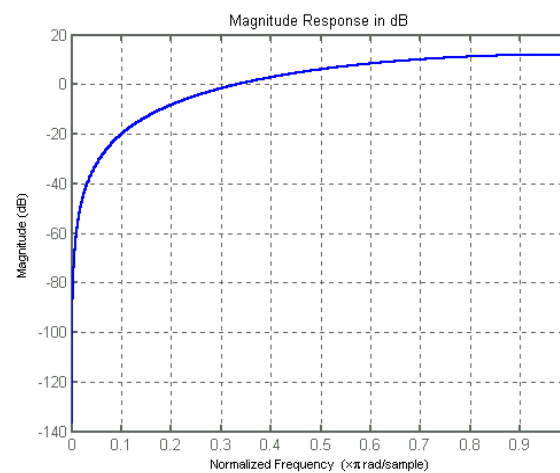


Figura 2-50 Respuesta en frecuencia (Magnitud) de un filtro de Noise - Shaping de orden 2

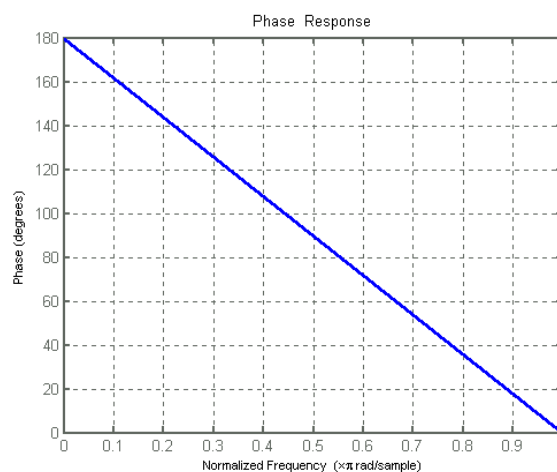


Figura 2-51 Respuesta en frecuencia (Fase) de un filtro de Noise - Shaping de orden 2

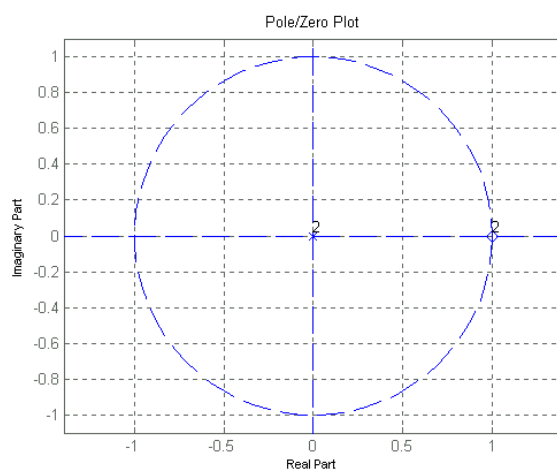


Figura 2-52 Diagrama de polos y ceros de un filtro de Noise-Shaping de orden 2

$n = 3$

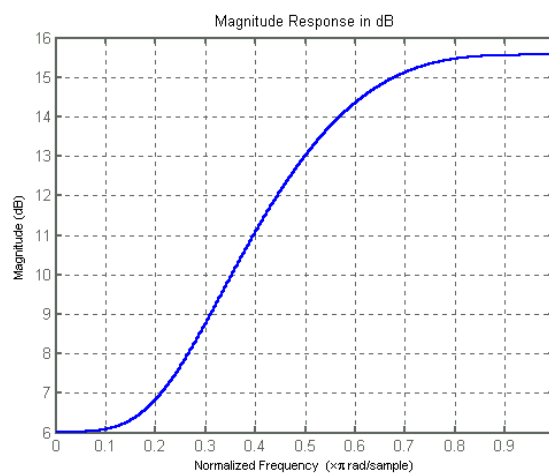


Figura 2-53 Respuesta en frecuencia (Magnitud) de un filtro de Noise - Shaping de orden 3

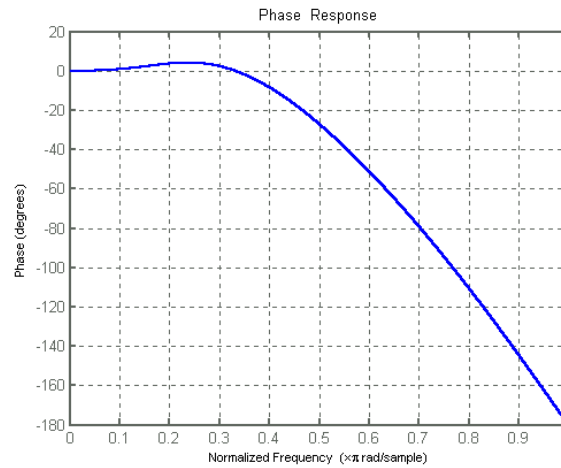


Figura 2-54 Respuesta en frecuencia (Fase) de un filtro de Noise - Shaping de orden 2

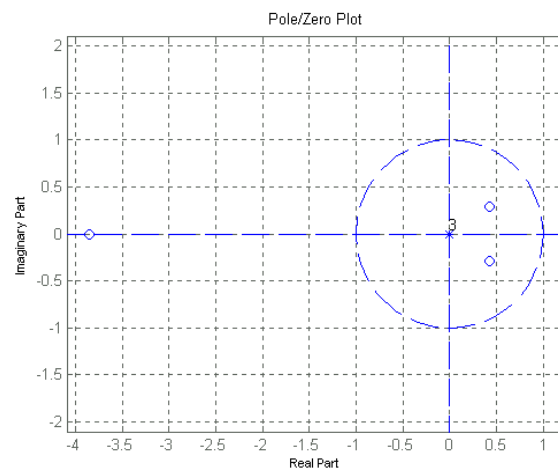


Figura 2-55 Diagrama de polos y ceros de un filtro de Noise-Shaping de orden 3

Respuesta en frecuencia comparativa

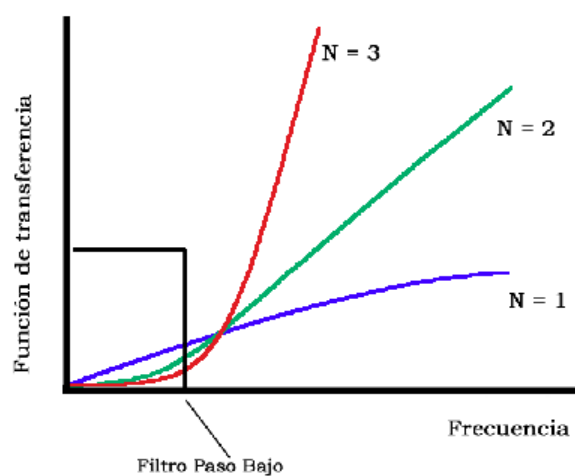


Figura 2-56 Comparativa de funciones de Noise - Shaping

Se observa en las gráficas anteriores como la cantidad de ruido desplazada es mayor con un orden más elevado, pero en ciertas zonas el filtro aplica una ganancia importante a tener en cuenta.

La distorsión de fase que introduce este tipo de filtros es también importante, como se puede ver.

Los diagramas de polos y ceros muestran que se trata de sistemas de estabilidad crítica, al estar los polos situados en cero.

A continuación, se presenta un gráfico comparativo del desplazamiento del error de cuantificación obtenido experimentalmente.

2.2.5.2 Estabilidad y ganancia

La estabilidad de un sistema es un factor fundamental a la hora de valorar su posible implementación práctica.

Los experimentos llevados a cabo con diferentes señales (sinusoidales, de voz, y de música) para estudiar su comportamiento (en Matlab y applets del tutorial) han demostrado que los filtros de *Noise-Shaping* de primer y segundo orden son estables en la gran mayoría de los casos. Por el contrario, los filtros de tercer orden y superiores presentan una gran inestabilidad y no son viables para ser implementados.

El efecto indeseado se produce cuando una muestra codificada por el sistema de *Noise-Shaping* debe tomar un valor que supera el fondo de escala, entonces se produce un recorte de la señal de salida y al entrar de nuevo en el lazo de realimentación, da lugar a resultados incoherentes. Esto se produce ya que el sistema introduce ganancia.

Estudio experimental

Se presenta un ejemplo que ilustra el fenómeno explicado.

Señal original de voz

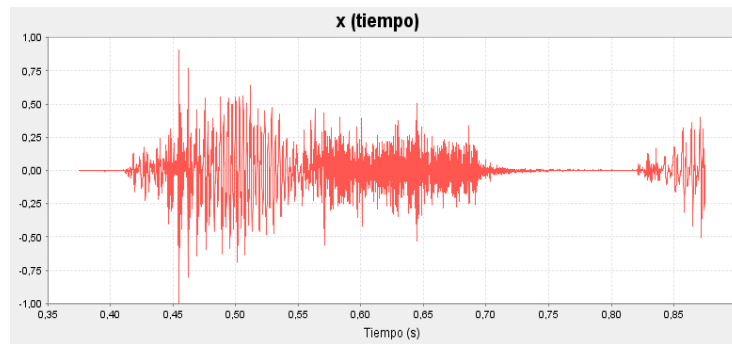


Figura 2-57 Señal de voz cuantificada con 16 bits

Señales obtenidas tras aplicar Noise Shaping (previo al filtrado paso bajo) realizando recuantificación a 4 bits (n : orden de la función de transferencia)

$n = 1$

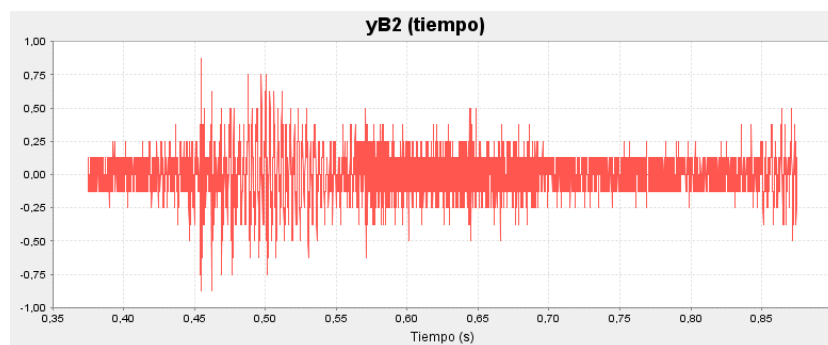


Figura 2-58 Señal de voz cuantificada con 4 bits aplicando Noise - Shaping de orden 1

$n = 2$

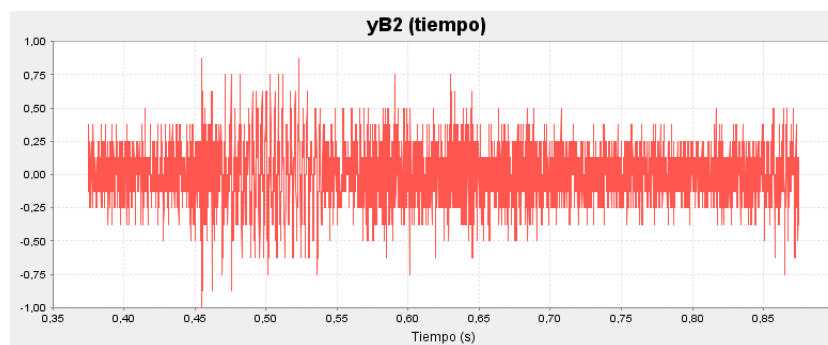


Figura 2-59 Señal de voz cuantificada con 4 bits aplicando Noise - Shaping de orden 2

$n = 3$

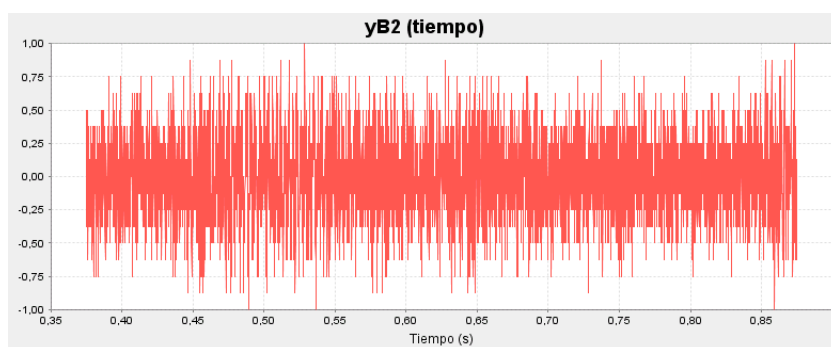


Figura 2-60 Señal de voz cuantificada con 4 bits aplicando Noise - Shaping de orden 3

$n = 4$

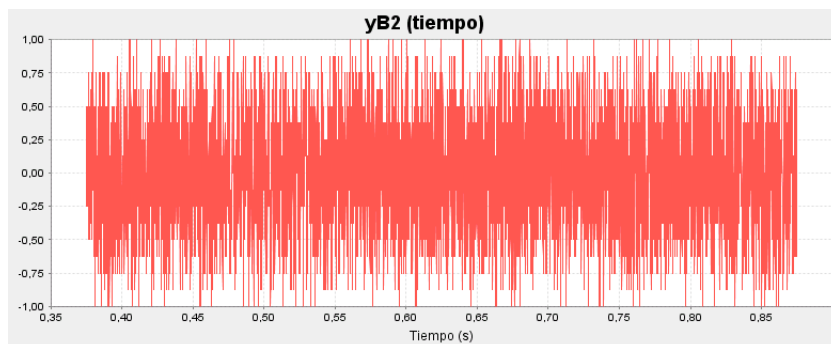


Figura 2-61 Señal de voz cuantificada con 4 bits aplicando Noise - Shaping de orden 4

El nivel de potencia del ruido de cuantificación se reduce a mayor orden utilizado: al pasar de primer a segundo orden, y de segundo al tercer orden.

En las gráficas se observa con claridad, como a mayor orden, la señal tiene una mayor ganancia, en el espectro, componentes de alta frecuencia que luego serán eliminadas por el filtro paso bajo del conversor digital-analógico.

En el caso del filtro Noise-Shaping de orden cuarto, sin embargo, la señal desborda el fondo de escala del cuantificador, aparecen muestras que salen fuera del rango $[1 \ -1]$, por lo que se produce recorte, esto provoca que el sistema se vuelva inestable.

En la siguiente tabla se aportan los resultados numéricos de la prueba anterior, en la que se aprecia, el cambio brusco al pasar de tercer al cuarto orden.

Tabla de nivel de potencia del ruido de cuantificación

Orden	Potencia ruido cuantificación
Sin aplicar Noise - Shaping	-30 dB
1	-36 dB
2	-41 dB
3	-42 dB
4	-25 dB

La estabilidad de los sistemas de Noise-Shaping ha sido objeto de investigación en numerosos estudios científicos, dada la importancia que tiene como se ha podido comprobar [22].

El criterio de estabilidad básico que se puede tomar como referencia es que:

"Para un filtro de Noise-Shaping de orden N , el número de bits del cuantificador debe ser igual o mayor que $N+1$ "

Esta condición se cumple en todos los casos estudiados en la aplicación construida (como el ejemplo expuesto anteriormente, para un cuantificador de 4 bits, el orden máximo de la función de transferencia que garantiza estabilidad es 3)

Los estudios realizados sobre la "Sigma-Delta Modulation" concluyen que es recomendable implementar filtros de primer o segundo orden como máximo. Esto verifica la condición anterior, puesto que en este caso, se trata de cuantificadores de un bit, para los cuales, la estabilidad es crítica.

2.2.5.3 Noise-Shaping basado en modelos psicoacústicos

La técnica de *Noise-Shaping* básica trata de desplazar la mayor cantidad de ruido de cuantificación posible a las altas frecuencias, con el fin de que se elimine una parte importante en combinación con la técnica de sobremuestreo, al quedar por encima de la frecuencia de corte; y que el ruido que, inevitablemente, quede dentro del ancho de banda de la señal, al estar situado en alta frecuencia, sea lo menos audible posible, conocida la baja sensibilidad del oído en esa zona.

Técnicas avanzadas plantean el método de conformar el ruido de cuantificación de acuerdo a las curvas de sensibilidad del oído humano, de modo que la mayor cantidad de ruido quede en las frecuencias donde la sensibilidad del oído es menor. La técnica presentada anteriormente, el *Noise-Shaping* tradicional, también se basa en el mismo concepto, solo que planteado de una forma muy idealizada, reduciéndose a considerar que el oído es menos sensible a mayor frecuencia, cosa que no es cierta de forma rigurosa.

Las curvas de audición (Equal – Loudness Curves)

El oído no presenta una respuesta en frecuencia plana, si no que responde de acuerdo a una función de transferencia que hace que ciertas frecuencias sean más audibles que otras.

La señal, desde que se recoge en el oído externo hasta que se percibe en la región del cerebro dedicada a la audición, sufre importantes modificaciones en su espectro, debido a su paso por diferentes canales (canal auditivo, oído medio, cóclea, procesamiento neuronal) así como en los procesos de transducción (acústico-mecánica, mecánica – acústica) [5] [8].

Este fenómeno hace que haya que diferenciar entre **intensidad de sonido** y **sonoridad**.

La **intensidad de sonido** es una magnitud medible que indica la potencia de un sonido en un punto dado. Se mide en decibelios SPL, con la fórmula:

$$SPL(dB) = 10 \log_{10} \left(\frac{P^2}{P_{ref}^2} \right)$$

La **sonoridad** es una magnitud subjetiva que indica la impresión auditiva que provoca cierto sonido sobre una persona. Se mide en “phons” y los valores correspondientes para cada nivel de intensidad sonora se han obtenido empíricamente.

Para cierto nivel de sonido, la sonoridad es mayor o menor según la frecuencia y según se trate de un tono puro o una banda de ruido.

Cada curva indica que para las frecuencias e intensidad que recorre, el oído percibirá la misma sonoridad. Así, un tono de 20 Hz de 75dB tiene la misma sonoridad que un tono de 1000 Hz de 10 dB: 10 phons.

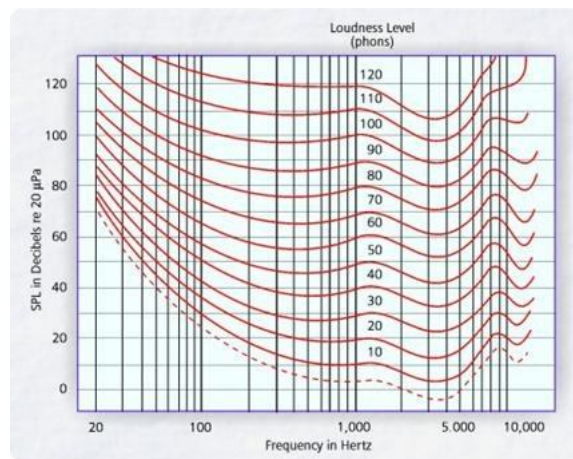


Figura 2-62 Curvas de audición (Equal - Loudness curves)

El rango de frecuencias que puede percibir el oído está comprendido entre 20 Hz y 20 KHz. La mayor sensibilidad se da en la franja entre 2 – 6 KHz. Para frecuencias altas o frecuencias muy bajas, la sensibilidad es menor.

El filtrado de la señal de sonido en el canal auditivo externo es uno de los principales factores que provocan la disposición de las curvas. En él se producen una serie de modos propios por la resonancia del sonido en dicha cavidad. Dado que tiene una longitud de 2.3 cm, los dos primeros modos propios se sitúan en 3.4 y 12 KHz, que como se observa en la figura, son las zonas de mayor realce.

El estudio de la fisiología de la audición se inició en 1933 con los trabajos de Fletcher y Munson, y tras décadas de perfeccionamiento, siendo destacables los avances de Robinson y Dadson, se han establecido una serie de curvas de audición,

recogidas en la norma ISO 266, que pueden ser utilizadas como estándar para cualquier trabajo de acústica.

Estas curvas se han obtenido sometiendo a grupos de personas a experimentos auditivos en condiciones controladas (cámara anecoica). Es importante resaltar el carácter empírico de este instrumento, que aún hoy, está sujeto a discusión y perfeccionamiento.

Es lógico, por tanto, concluir que para aplicar *Noise-Shaping* en un proceso de cuantificación, lo más óptimo será desplazar el ruido a las frecuencias donde el oído es menos sensible, eliminándolo de aquellas donde es más sensible, utilizando las curvas *Equal – Loudness* como referencia, en vez de limitarse, simplemente, a reducir, por igual, el ruido de cuantificación en toda la banda audible (20 Hz – 20 KHz).

Diseño e implementación

El diseño de la función de transferencia de un sistema de Noise – Shaping basado en un modelo psicoacústico es una tarea compleja que se ha realizado con diferentes aproximaciones. El objetivo es minimizar N_w , la potencia de ruido “percibida”, es decir, la potencia de ruido de cuantificación ponderada con un modelo psicoacústico:

$$N_w = \int_{-\pi}^{\pi} P_{E'}(e^{j\omega}) W(\omega) d(\omega)$$

$P_{E'}$ es la potencia de ruido resultante de aplicar el filtro de *Noise – Shaping*:

$$P_{E'}(e^{j\omega}) = \|1 - H(e^{j\omega})\|^2 P_E(e^{j\omega})$$

El cálculo de los coeficientes del filtro H no es trivial y se han propuesto diversos métodos matemáticos para lograrlo.

Un factor importante a considerar es que el cálculo de los coeficientes sea fijo, es decir, independiente de la señal de entrada o dinámico, es decir, que se calculen en tiempo real en función de la señal de entrada.

Métodos numéricos

El sistema de ecuaciones se puede resolver mediante métodos numéricos [16], pero los algoritmos diseñados requieren un gran coste computacional, con lo que tienen el inconveniente de que no es viable su cálculo en tiempo real. Dan lugar, por tanto, a una solución cerrada, una serie de coeficientes fijos que definen una función de transferencia que no depende de la señal de entrada, con lo cual, el modelo psicoacústico no es el más óptimo posible.

Técnica de los mínimos cuadrados

La técnica de los mínimos cuadrados propone un algoritmo para calcular los coeficientes del filtro, utilizando un sistema de ecuaciones normales, con la suficiente simplicidad para que se pueda realizar en tiempo real en función de la señal de entrada [9].

Se define, de la manera habitual, el filtro IIR de *Noise-Shaping* como un polinomio de la forma:

$$1 - H(z) = \sum_{n=0}^M a(n) z^{-n}$$

La función a minimizar se expresa en términos del filtro IIR de *Noise-Shaping*:

$$E_{NS} = \int_{-\pi}^{\pi} \left\| \sum_{n=0}^M a(n) e^{-j\omega n} \right\|^2 P_E(e^{j\omega}) W(\omega) d\omega$$

Aplicando el teorema de Parseval se transforma la ecuación anterior al dominio del tiempo, en el cual, se podrá trabajar para encontrar los coeficientes del filtro.

La función $W(\omega)$ se transforma en $V(\omega)$ como $V(\omega) = \sqrt{W(\omega)}$, de modo que el problema se transforma en un problema de mínimos cuadrados que se puede resolver

$$E_{NS} = \sum_{n=0}^{\infty} \left(\sum_{k=0}^M a(k)v(n-k) \right)^2$$

El objetivo es encontrar los coeficientes a_k que minimizan E_{NS} , luego se deriva la función respecto a los a_k y se iguala a cero

$$\frac{\partial E_{NS}}{\partial a(k)} = 0, k = 1 \dots M$$

Esto da lugar a las ecuaciones normales

$$Ra = r$$

r es la función de autocorrelación, que se calcula como la Transformada de Fourier inversa de la función perceptual $W(w)$

Los coeficientes a_k se pueden despejar por diversos métodos.

Función de modelado psicoacústico

Existen varias posibilidades en la elección de la función de modelado psicoacústico que aplicar, basadas en las curvas *Equal – Loudness* que se han comentado.

En general, las curvas F [2] y la E [10] son las más habituales y de uso común.

La función que se aplica en el sistema de *Noise-Shaping* es la inversa de la función definida, el objetivo es atenuar al máximo el ruido en aquellas frecuencias donde las curvas indican que el oído es más sensible.

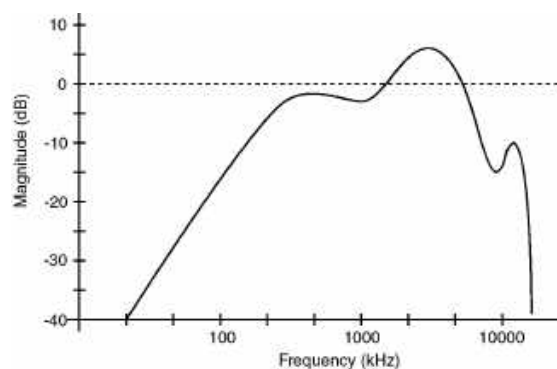


Figura 2-63 E-weighting curve

Se tratan de funciones fijas que no varían en función de la señal de entrada.

Existen otras funciones más avanzadas, en las que la ésta depende de la señal de entrada (P_{xx}) y un modelo psicoacústico (P_{TQ}) [9].

$$W(w_k) = \frac{1}{\beta P_{xx}(w_k) + (1 - \beta) P_{TQ}(w_k)}$$

Al tener que calcularse de forma periódica, sobre partes de señal enventanada, el coste computacional de calcular W debe ser bajo.



TECNOLOGÍAS DE DESARROLLO

3 Tecnologías de desarrollo

3.1 El entorno web

3.1.1 Introducción

El entorno web es un conjunto tecnologías de presentación de contenidos multimedia con una extraordinaria popularidad en nuestros días, gracias al proceso en el cual, desde mediados de los años noventa, el acceso a la red Internet se ha difundido rápidamente a toda la sociedad, impulsada por los avances técnicos que han mejorado su potencial (como el ADSL) y el abaratamiento de su coste. La World Wide Web ha sido el impulso fundamental de la llamada “Sociedad de la Información”.

Por la facilidad para conectarse a ella (el usuario podrá hacerlo desde un ordenador de una aula informática de su universidad o centro de estudios, desde su casa con conexión ADSL, o con un portátil en una zona WiFi), por la sencillez de uso, pues tan solo es necesario que el usuario sepa manejar un navegador como Firefox o Explorer y por el potencial para distribuir el producto, pues tan sólo habrá que “colgarlo” en un servidor como por ejemplo, el de la universidad, para tener una audiencia potencial de millones de personas, el entorno web es un medio ideal para desarrollar el proyecto.

Aunque estas razones puedan parecer obvias e incluso, redundantes respecto a lo que ya se ha comentado en capítulos anteriores, no lo es tanto y cabe insistir en señalarlas. Hace solo unos pocos años, a finales de los noventa, hubiera sido impensable desarrollar una solución de este tipo, pues los primeros navegadores, como Explorer o Netscape, ofrecían posibilidades muy limitadas de interactividad.

Los estándares web reconocidos y regulados por la máxima autoridad mundial en la materia, el W3C, son XHTML, CSS y Javascript. A partir de ahora, se denominará “documento web” al resultado de crear un proyecto basado en estas tecnologías (coloquialmente conocido como “página web”).

Hay que mencionar que en los documentos web se pueden introducir objetos que se han incorporado para aumentar la funcionalidad y las posibilidades

de la web, desarrollados por empresas privadas, como el popular formato de animaciones Flash de Adobe. En este proyecto, se evitará introducir este tipo de soluciones propietarias puesto que no ofrecen ventajas significativas y conllevan los evidentes problemas asociados a la utilización de tecnologías no estándares, como la dependencia de un propietario, obligar al usuario a instalar o actualizar plugins, etc. Los estándares del W3C son suficientes para crear cualquier tipo de elemento, incluso animaciones complejas.

XHTML es un lenguaje de etiquetas destinado a crear la estructura básica del documento; derivado de HTML, prescinde de toda la información de estilo que ofrecía HTML, para realizar documentos web con total separación entre contenido y estilo de presentación. CSS es el componente que se encarga de establecer, precisamente, el estilo de presentación. Javascript, finalmente, se encarga de aportar interactividad al documento. La combinación de estas tres tecnologías tiene un poderoso potencial, la separación entre contenido y presentación es su gran fortaleza.

Las ventajas de este modo de trabajar son notables. Las páginas web antiguas, que no utilizaban CSS, llevaban los parámetros de estilo embebidos dentro del contenido, lo cual, plantea múltiples dificultades a la hora de desarrollar y sobretodo, mantener un sitio web. A la vez que se introduce el contenido, deben introducirse etiquetas de estilo, por ejemplo, tipo de fuente, tamaño, márgenes, colores. (, , <color>). Si pasado un tiempo, se desea cambiar el aspecto de la página, debe reescribirse todo el código, lo cual es laborioso y costoso. Con la combinación de XHTML y CSS, en cambio, tan solo se marcan las distintas partes del contenido según su tipo, por ejemplo, párrafo, título, imagen (<p>, <h1>, <h2>...<h8>,) y en el archivo CSS asociado, se establece el aspecto de cada uno, pudiéndose incluso, crear subtipos de cada elemento mediante el atributo *class*. Si se desea cambiar el estilo de presentación de la página o de varias páginas, tan sólo deben modificarse los atributos CSS especificados para cada elemento y esto afectará globalmente a toda la página web.

A continuación, se detallan brevemente las particularidades de cada uno de los componentes.

3.1.2 Lenguajes de presentación

3.1.2.1 XHTML

XHTML, como se ha dicho, es un lenguaje de marcas que tiene como fin crear la estructura básica del documento web. Se trata de una versión de HTML 4.0 que se atiene a ciertas normas para crear un documento limpio y bien estructurado, corrigiendo la flexibilidad de HTML [32].

HTML es un lenguaje de etiquetado que surgió a principios de los años noventa como estándar para la creación de páginas web de Internet, impulsado por Tim Berners-Lee. Los documentos HTML se comienzan a escribir insertando la marca `<html>` y descendiendo de ella, dos marcas principales, `<head>`, para insertar metadatos y ficheros relacionados con la página, y `<body>`, a partir de la cual se crea la estructura del documento, a base de combinar títulos `<h1>`, párrafos `<p>`, imágenes `` y otros elementos más complejos, como tablas, listas, etc. Estos elementos se pueden agrupar en cajas `<div>` para maquetar visualmente la página. Para establecer las dimensiones y aspecto de cada elemento se introducen marcas de estilo. Todas las etiquetas se cierran para estructurar el documento.

Durante los años noventa fue mejorado progresivamente, introduciéndose nuevas etiquetas, hasta llegar la versión 4. Una de las características más relevantes de HTML es su flexibilidad en la sintaxis, para que un navegador represente un documento web no es necesario que todas las etiquetas estén cerradas e incluso pueden existir incoherencias, aunque el resultado, obviamente, será deficiente.

Con la difusión de Internet y la aparición de nuevas y mejores versiones de los navegadores, se empezaron a crear páginas web cada vez más complejas. De forma paralela, surgió el lenguaje XML, para organizar datos de forma estructurada, es decir, jerárquica, también mediante etiquetas. Los documentos XML definen la estructura de datos mediante un documento opcional, la DTD, que establece como pueden combinarse las etiquetas, cuales son obligatorias, etc, en definitiva, como deben introducirse los datos, para que todas las personas que tengan que crear un documento XML de un determinado tipo sigan las mismas reglas y todos tengan la misma estructura.

XHTML es, entonces, una versión de HTML que se ajusta a una DTD definida por el W3C que especifica ciertas reglas (en realidad, existen varias versiones, cada una definida con su DTD correspondiente). Con XHTML, se eliminaron todos los problemas que acarreaba la laxitud en la creación de documentos HTML, como la no obligación de cerrar etiquetas, y lo que es un avance fundamental, se eliminaron todas las etiquetas de estilo, obligándose a utilizar CSS, impulsando una de las buenas prácticas más importantes recomendadas por el W3C, la separación entre contenido y presentación.

DTD's de XHTML

El WC3 ha definido tres DTD's con las que crear documentos XHTML. Cada una de ellas define un conjunto de reglas para crear el documento, varían en la rigurosidad que se impone a la hora de seguir las normas de sintaxis, elementos y atributos permitidos, etc. La DTD utilizada se indica con una etiqueta DOCTYPE insertada en el comienzo del documento, como en cualquier documento XML.

Strict

La DTD Strict define el documento XHTML más riguroso, en el cual, se debe seguir la sintaxis correcta (cerrar etiquetas, minúsculas, etc) y además, no se permite utilizar ningún tipo de etiqueta de estilo.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

Transitional

La DTD Transitional exige tan sólo que el documento creado esté bien formado, es decir, que siga las reglas de sintaxis de XML, estando permitidas ciertas etiquetas de estilo

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

Frameset

La DTD Frameset permite el uso de frames, elemento HTML prohibido por las otras DTD's que todavía tiene un uso bastante extendido.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

Frames

Las frames son un elemento especial de HTML. Consisten en una etiqueta que se introduce en el documento HTML para mostrar el contenido de otra página web, interna o externa. Es decir, se trata de visualizar de forma embebida una página html dentro del *body* de otra, separada del resto del contenido mediante bordes.

Existen dos tipos de frames. Las clásicas, cuya etiqueta era <frame> y que para ser insertadas, debe definirse un contenedor principal en el cual, irán insertadas dos o más frames ocupando todo el espacio y las modernas, denominadas "iframe", que se pueden introducir del mismo modo que una caja div, en cualquier lugar del body del documento contenedor, mezclada con elementos como imágenes, div's, párrafos, etc.

El uso de frames (o iframes) presenta una serie de ventajas e inconvenientes que deben ser analizados para evaluar su posible uso en el proyecto. Se trata de un elemento controvertido, sobre el cual es discutido si tienen un uso justificado o debe abandonarse paulatinamente; de hecho, como se ha visto, se ha creado una especificación de DTD concreta para albergar este elemento, puesto que en el caso estricto, XHTML no considera su existencia.

3.1.2.2 CSS

CSS es un método para definir la presentación y el aspecto visual de los documentos web. Se trata de un documento con una lista de atributos que establecen las propiedades de los elementos que forman el archivo html, así, por ejemplo, a los párrafos se les puede asignar tipo, tamaño y color de la fuente con las propiedades font-family, font-size y color respectivamente, a las cajas div o al body se les puede definir sus dimensiones, color o imagen de fondo, con los atributos width, height, background-color o background-image por ejemplo [33].

Su denominación, hojas de estilo en cascada, se refiere al funcionamiento a base de una cascada o jerarquía de las hojas CSS en conjunto con el documento xml.

Mediante un selector se determina el elemento del que se quiere establecer sus propiedades, y entre corchetes, se inserta la lista de los atributos con su valor correspondiente, por ejemplo:

```
p {  
  font-family:Arial;  
  font-size:12pt;  
  color:blue;  
}
```

Insertando este bloque en el CSS asociado al documento todos los párrafos se visualizarán con fuente tipo Arial, de tamaño 12 puntos y color azul.

Para maquetar documentos web complejos en los que existan varios tipos del mismo elemento, se utiliza el atributo *class* en las etiquetas HTML. Este atributo sirve para crear grupos de elementos que tendrán las mismas propiedades CSS y en éste, se determinarán dichas propiedades mediante un selector formado por ".valor_atributo". Por ejemplo, para crear dos tipos de párrafos diferentes:

Documento HTML

```
<html>  
<body>  
  <p class="tipoA">Lorem Ipsum</p>  
  <p class="tipoB">Lorem Ipsum</p>  
</body>  
</html>
```

Hoja de estilo CSS asociada

```
p.tipoA{
font-family:Arial;
font-size:12pt;
color:black;
}
p.tipoB{
font-family:Times;
font-size:10pt;
color:blue;
font-style:italic;
}
```

La característica de que las hojas de estilo se estructuren en cascada permite que se puedan establecer atributos globales para los contenedores de grupos de elementos y luego, para cada uno de los elementos que los forman, refinar las propiedades que sean necesarias. Así, por ejemplo, si el color de fuente habitual de la página es el negro, puede introducirse `color:black` en el selector del `body`, por lo que se aplicará por defecto a todos sus elementos hijos, sean del tipo que sean (párrafos, títulos, etc) pero si existe algún caso especial, se introducirá el atributo `color` en su selector con el valor deseado, sobrescribiéndose el valor que recibiría por defecto.

Existen tres maneras diferentes de asociar la hoja de estilo al documento `html`. Se pueden insertar los atributos directamente dentro de cada etiqueta con el atributo `style`; se puede introducir el contenido de la hoja de estilo dentro una etiqueta `style` insertada en el `head`, o bien, crear un fichero `.css` en el que se guardará la hoja de estilo y que será asociado al archivo `html` mediante una etiqueta `link` en `head`.

El último método es el más poderoso y con el cual se puede aprovechar al máximo las posibilidades de separación de contenido y estilo del sistema `XHTML+CSS`. Si se tiene un conjunto de documentos `html` que mediante hiperenlaces entre ellos forman un sitio web, se puede aplicar la misma hoja de estilo a todos, ahorrando información redundante pero lo que es más importante, se puede cambiar el estilo de todo el sitio web con tan sólo modificar el archivo

CSS, los cambios afectarán a todas las páginas que lo forman, sin necesidad de reescribir una a una todas ellas.

CSS 2.0 es actualmente la versión más extendida e implementada del sistema, la versión 3.0 se encuentra en fase de estudio y desarrollo.

3.1.2.3 *Javascript*

Javascript es un lenguaje de programación que se utiliza para dar funcionalidad dinámica a las páginas web. Es un lenguaje interpretado, es decir, no requiere compilación y provee la capacidad de crear objetos, si bien, de forma muy rudimentaria.

Apareció en el navegador Netscape 2.0 y tuvo inicialmente los nombres Mocha y LiveScript, la versión definitiva fue presentada por Netscape y Sun Microsystems en 1995. Debido a su gran difusión acabó convirtiéndose en un estándar de ECMA (denominándose ECMAScript) en 1997 y posteriormente, de ISO.

Alcanzó una gran popularidad en los años noventa pues era el único método existente en la época para crear webs dinámicas. Posteriormente, perdería cuota de mercado debido a la aparición de formatos como Flash, destinados a crear animaciones y efectos con gran facilidad mediante entornos de desarrollo como Dreamweaver, si bien, como se ha explicado, se trata de soluciones propietarias. En los últimos tiempos Javascript está volviendo a adquirir popularidad impulsado por AJAX, un sistema que facilita la creación de páginas web dinámicas sincronizadas con el servidor, gracias a la inclusión del objeto XMLHttpRequest.

Es importante insistir en que con Javascript se pueden lograr los mismos efectos, de la misma espectacularidad, que con las películas Flash, la razón por la que este último ha sido elegido por un gran número de desarrolladores es la sencillez con la que se pueden diseñar los efectos, a través de una interfaz gráfica. No obstante, Javascript, cuenta con la ventaja de que existe una gran variedad de frameworks como JQuery, Prototype o Yahoo User Interface que facilitan la tarea de crear interfaces y elementos complejos [35].

El mecanismo básico para implementar funcionalidades con Javascript es programar funciones que responden a eventos asignados al elemento correspondiente de la interfaz gráfica. Dichas funciones pueden implementarse de diversas maneras, al igual que sucedía con CSS, insertándose en la propia página html o enlazando un archivo externo.

Es importante mencionar que una de las claves del funcionamiento de Javascript es el árbol DOM. Éste es un modelo estandarizado que define el acceso y la interacción del script con documentos de marcas, como XML o XHTML. Con él, se puede acceder a los diferentes elementos que componen la estructura del documento y modificarlos. Se pueden añadir o eliminar elementos, modificar dinámicamente sus propiedades CSS, etc.

3.1.3 Librería de Javascript JQuery

JQuery es un framework de JavaScript cuyo objetivo es aumentar la productividad a la hora de crear aplicaciones web. Contiene una serie de utilidades que encapsulan las tareas más frecuentes que se suelen necesitarse para implementar sitios web, en lo referente a la parte del cliente (la vista), las cuales, en general, son tediosas y requieren una gran carga de código JavaScript, por lo que se simplifica en extremo el desarrollo de la parte lógica del cliente.

Las funciones que ofrece están relacionadas con la manipulación del árbol DOM del documento web, la modificación dinámica de la hoja de estilo CSS, la captura y el tratamiento de eventos generados por el usuario, y la creación de animaciones sobre elementos gráficos del sitio web. Todas ellas, además, con la gran ventaja de que encapsulan, de forma transparente para el desarrollador web, el tratamiento específico de cada operación para diferentes navegadores (Firefox, Internet Explorer, etc), por lo que facilita enormemente la compatibilidad entre navegadores, y el desarrollo puede realizarse sin tener en cuenta las particularidades de JavaScript y CSS para cada navegador, lo cual es especialmente útil para el desarrollo específico para Internet Explorer, que es el que más diferencias tiene.

La necesidad de JavaScript en el tutorial es elemental, y tan solo se recurre a él para gestionar la navegación por las diferentes áreas del mismo, así como el

control sobre ciertas propiedades CSS que varían en función de la navegación. Aún así, teniendo en cuenta la sencillez del objetivo que se persigue, la potencia de esta librería es tal que su uso es una importante ayuda, simplificándose notablemente la implementación de esta parte del desarrollo. Por estos motivos, se ha seleccionado esta librería como herramienta de apoyo en la creación del script que maneja la web del proyecto.

Como única desventaja reseñable se puede mencionar el hecho de aumentar el peso de la descarga que el navegador tiene que llevar a cabo para presentar la web del tutorial, puesto que el uso de JQuery requiere añadir el fichero js que contiene la librería. Se trata, sin embargo, de un fichero liviano, de apenas 120 kB, por lo que dadas las prestaciones actuales de las líneas de conexión a Internet, se trata de un añadido insignificante. La implementación directa sobre JavaScript “a bajo nivel” aunque evitaría tener que incluir este fichero, aumentaría el tiempo y la complejidad del desarrollo y la probabilidad de errores dada la problemática que presenta este lenguaje como se ha mencionado antes.

3.1.4 Cuestiones de diseño web

3.1.4.1 Desventajas y puntos débiles del entorno web

El entorno web presenta una serie de inconvenientes y problemática, lo que debe ser tenido en cuenta a la hora de plantear el proyecto y desarrollar la solución.

El principal inconveniente es la dependencia con el navegador. Si bien, el navegador es en principio uno de los motivos por los cuales se logra el requisito del funcionamiento multiplataforma, en última instancia, es el navegador, como es lógico, el que presenta los contenidos y por tanto, se depende de él.

Debido a la amplia variedad de navegadores que existen en el mercado, y sobre todo, por el hecho de que no todos cumplen los estándares, existe un grave riesgo de que la solución desarrollada no se presente y funcione de igual manera en todas las plataformas. La compatibilidad entre navegadores es uno de los caballos de batalla en el campo del desarrollo y la mejora de la Web. A continuación, se expone la situación actual del mercado de los navegadores y como será tomada en cuenta para el desarrollo del proyecto.

3.1.4.2 Navegadores de Internet

El tutorial de digitalización de audio será accesible por Internet al tratarse de un sitio web. Por tanto, la plataforma básica para ejecutarlo será un navegador web. Como se ha explicado, se trata de un elemento que debe ser considerado cuidadosamente antes de proceder al desarrollo del diseño web, ya que es fuente habitual de problemas e incompatibilidades. En el mercado existe una gran variedad de ellos, de muy diferentes características, cada uno con sus particularidades, por este motivo, crear un sitio web perfectamente usable desde cualquier navegador es uno de los problemas fundamentales del diseño web.

El navegador con mayor popularidad actualmente es Internet Explorer, de Microsoft (actualmente en su versión 7), con una cuota de mercado alrededor del 80%. Es con diferencia el más utilizado, aunque su calidad sea discutida. Tras él, se encuentra Firefox (actualmente en su versión 3) con una cuota de mercado del 10% aproximadamente. Se trata de software libre desarrollado por una comunidad de programadores (Mozilla). Es un navegador especialmente útil para el desarrollo web, pues consta de una serie de herramientas que facilitan la tarea del maquetado CSS, depuración de Javascript, etc, como por ejemplo, Firebug. Finalmente el resto del mercado se lo reparten una serie de navegadores como Safari, Ópera y Google Chrome, este último de reciente aparición y del que se prevé un enorme potencial.

La parte principal de la problemática se debe a la interpretación de las reglas CSS que hacen los navegadores. Dado que no todos dan el mismo resultado, es decir, no presentan los elementos de igual manera para una misma hoja de estilo CSS, maquetar una página de modo que se vea exactamente igual en todos es una tarea complicada. Si la web es muy compleja, puede llegar a ser necesario crear una hoja CSS específica para cada uno, ya que entre las funcionalidades que Javascript ofrece, está la de identificar el agente (navegador) que está mostrando la página. No obstante, a pesar de la heterogeneidad del mercado de los navegadores, hay que decir que al final, todos se basan en uno de los tres principales motores de renderización HTML que existen: Gecko (utilizado por Firefox), Webkit (utilizado por Safari y Chrome) y Trident (utilizando por Explorer).

En la práctica, la mayor parte de los problemas surgen con el navegador Explorer, pues es el que implementa de una manera menos rigurosa las reglas CSS

y las normas de acceso al árbol DOM. Adaptar las páginas para este navegador suele ser complicado, sin embargo, como se ha dicho, es el de mayor difusión, por lo que habitualmente se hace imprescindible, en cualquier desarrollo profesional de una web, realizar el esfuerzo de conseguir la compatibilidad con Explorer.

Existen diversos métodos para comprobar de forma precisa la calidad de los navegadores, más allá de impresiones subjetivas o prejuicios asentados. El más popular es el test Acid. Se trata de un sitio web en el que se construye un dibujo a base de combinar elementos HTML con las propiedades CSS correspondientes. Si el navegador implementa CSS de forma rigurosa, debería mostrarla correctamente, sin embargo, ya que en la práctica no es así, aparecen errores. A continuación, se presenta un ejemplo de aplicación del test Acid en Firefox y Explorer

Hello World!

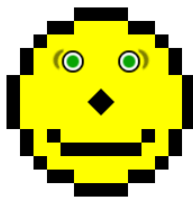


Figura 3-1 Test Acid para Firefox 3

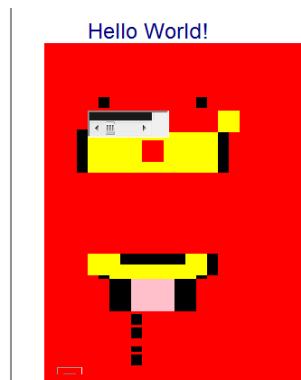


Figura 3-2 Test Acid para Internet Explorer 7

La diferencia entre el renderizado de Firefox y el de Explorer es evidente. No obstante, no debe tomarse este test como excusa para desatender el navegador Explorer en el desarrollo web, pues es el utilizado por un mayor número de usuarios. Hay que tener en cuenta que se trata de un test extremo, un caso muy

particular, pero que en la mayoría de casos de diseño habituales, las diferencias serán solo detalles que podrán ser resueltos sin excesiva dificultad.

En cuanto a las implicaciones de esta problemática para el proyecto, cabe decir que uno de los requisitos fundamentales del mismo es que el producto final sea multiplataforma, es decir, ejecutable en cualquier entorno (sistema operativo o navegador). Por tanto, el diseño web deberá realizarse orientado a obtener un sitio web compatible con cualquiera de los navegadores más utilizados actualmente: Firefox, Safari, Opera e Internet Explorer, a pesar de que este último pueda presentar alguna dificultad. Dado que la complejidad del proyecto está, esencialmente, en el desarrollo de las applets, y el entorno web consistirá en una combinación de texto e imágenes con efectos visuales sencillos, con una interactividad basada principalmente en el hipertexto, no es éste un problema crucial.

No obstante, no se tendrá como objetivo conseguir una compatibilidad perfecta con todos los navegadores. Ciertas irregularidades podrán ser despreciables mientras el tutorial sea usable. Por este motivo, en el inicio del tutorial, se recomendará el navegador más indicado, aunque no sea un impedimento utilizar otro.

Por tanto, por los motivos citados, el proyecto será creado sobre Firefox (por la variedad de herramientas de desarrollo que ofrece) por lo que entonces, será el navegador que en la página de inicio y en la ayuda se recomendará usar. Los navegadores Safari, Ópera y Google Chrome pueden ser considerados análogos a Firefox puesto que respetan las recomendaciones del W3C con la misma rigurosidad, por lo que es presumible obtener unos resultados también satisfactorios. En cuanto a Explorer, se realizará el pertinente trabajo de adaptación a él, para que sea accesible a los usuarios del mismo, aunque se permitirá que existan ciertas diferencias despreciables.

3.1.4.3 *Diseño web y usabilidad*

Los conceptos expuestos hasta ahora han sido fundamentalmente técnicos y definen las tecnologías y herramientas que posibilitarán la implementación de un tutorial en una plataforma web. Sin embargo, en este punto, surge la necesidad de

decidir cómo utilizar estas tecnologías. Además de comprender su funcionamiento y entender de qué manera construir la arquitectura web, es necesario estudiar de qué modo se presentará la información para lograr de una manera efectiva el objetivo del proyecto, que no es otro que crear una herramienta didáctica que ayude a los alumnos a comprender mejor la materia de la asignatura. Aunque se utilicen las tecnologías más avanzadas, aunque desde un punto de vista técnico el diseño de la arquitectura web sea la más óptima, si los conceptos de presentación de información y usabilidad se descuidan, el resultado será un sitio web que no cumple con su cometido, en este caso, didáctico.

Hay que señalar, no obstante, que los requerimientos del proyecto respecto a esta cuestión son elementales. El diseño de un sitio web profesional, con fines de negocio, puede tener cientos de opciones, una estructura de navegación muy compleja, con varios niveles de profundidad, múltiples formularios, presentación de gran cantidad de datos. Es por ello que la usabilidad y el diseño de la interacción humana con las páginas web son toda una disciplina y su aplicación no es algo trivial. Sin embargo, las necesidades de este proyecto son limitadas. El tutorial carece de formularios, no es necesaria la interacción con un servidor (creación de un modelo MVC) y el número áreas temáticas relativamente pequeño. Por tanto, se tomarán en consideración algunas pautas elementales para la creación de un diseño web que asegure la claridad en la presentación de la información, esencial para una web con fines didácticos y que facilite la navegación [35].

3.1.4.3.1 Principios de diseño web

Se presentan una serie de buenas prácticas encaminadas a construir sitios web con calidad y eficacia. Se trata objetivos que es deseable que la web los cumpla en la mayor medida posible [36].

- Anticipación: el sitio web debe de prever los objetivos del usuario y por tanto, mostrar las opciones de manera que faciliten su consecución, no que la entorpezcan.
- Autonomía: el usuario debe de tener un control total, en todo momento, de la navegación y el acceso a los contenidos del sitio web, en ningún momento puede sentirse perdido.

- Eficiencia: el sitio web debe proporcionar una navegación eficaz, facilitar al usuario el acceso a la información que busca de la forma más rápida posible, teniendo en cuenta siempre que su tiempo es muy valioso.
- Aprendizaje: se requerirá cierto tiempo para que el usuario se familiarice con el sitio web y su navegación, pero éste debe ser reducido al mínimo posible.
- Legibilidad: debe de haber contraste entre el texto y el fondo para facilitar su lectura y el tamaño de la fuente, lo suficientemente grande.
- Interfaz visible: deben evitarse los elementos ocultos o que necesiten ser desplegados mediante una acción del usuario, especialmente, si no son evidentes.
- Maximizar la relación entre el esfuerzo del usuario y los objetivos alcanzados
- Limitar el número de opciones al mínimo necesario, si se presentan demasiadas posibilidades en pantalla, y muchas de ellas no son imprescindibles en el contexto en el que se encuentra el usuario, se aumenta peligrosamente la posibilidad de error.
- Sobrecarga de información: no debe presentarse una cantidad de información excesiva. El usuario tiene que saber, con facilidad, de un solo vistazo, dónde empezar a leer y cómo continuar una vez que ha finalizado con un apartado.
- Consistencia: el sitio web debe mantener una estructura homogénea y coherente durante toda la navegación en él. La presentación de contenidos del mismo tipo debe realizarse siempre de la misma manera, para facilitar al usuario familiarizarse con el sitio.

3.1.4.3.2 Diseño de links

Debe mantenerse, en medida de lo posible, el formato tradicional de link azul y subrayado. Aunque se hayan popularizado otras opciones estéticamente mucho más vistosas, como botones con degradados de color, los usuarios están acostumbrados a este modelo tradicional, y pueden sentirse perdidos si no se respeta [36].

3.1.4.3.3 Diseño de menús

Los menús son el elemento de navegación más popular y de mayor uso en los sitios web, sin embargo, es importante utilizarlos con racionalidad y moderación. En primer lugar, hay que clasificar los contenidos de la página web de forma concisa, agrupándolos en áreas temáticas muy generales que faciliten al usuario orientarse, evitando presentar largos listados de capítulos. La idea fundamental es que el menú principal, y cualquier submenú derivado, no presente más de siete opciones. Hay que evitar en medida de lo posible las opciones desplegables, presentando todo el contenido de forma directa, siempre que esto no suponga una sobrecarga de información. En caso de que sea imprescindible, debe indicarse de una manera explícita y clara el despliegue de las opciones, dejando una, al menos, abierta por defecto, cuando se carga la página web [36].

3.1.4.3.4 Uso de frames

El uso de iFrames es una opción de diseño que compete al desarrollador web. Se desaconseja su uso de manera si no es imprescindible, pero para ciertos casos, es la solución más eficaz. En cualquier caso, los iFrames, tienen notables ventajas sobre los frames tradicionales que hacen viable su utilización, por ejemplo, no es necesario establecer un HTML base que contenga al conjunto de frames, si no que se pueden insertar en cualquier lugar mediante posicionamiento por CSS [36].

3.1.4.3.5 El espacio de la pantalla

Es necesario distribuir las diferentes áreas de información presentadas en la pantalla separándolas mediante espacios vacíos que faciliten al usuario la visualización. Aún cuando la interfaz sea extremadamente simple como, por ejemplo, un índice y una página, es necesario dejar un espacio adecuado de márgenes a los lados y entre los elementos. Según el experto Jakob Nielsen, el contenido de la página debe ocupar entre un 50% y 80 %, el resto, se debe dedicar a márgenes y espacio vacío. Es erróneo introducir la mayor cantidad de elementos posibles en una sola pantalla, para dar sensación de riqueza de información o para hacer que el usuario vea la mayor cantidad de opciones posible, sin necesidad de navegar, pues causaría confusión y un esfuerzo adicional. En cambio, hay que

simplificar en medida de lo posible cada página que compone un sitio web, de modo que sea fácil enfocar la atención del usuario. Los elementos gráficos de navegación hay que reducirlos al mínimo, tanto en número como en espacio ocupado, recordando que son una herramienta para navegar por el sitio web, no un fin en sí mismos [36].

3.1.4.3.6 Columnas y legibilidad del texto

La correcta presentación del texto en la pantalla, que facilite la legibilidad del mismo, es esencial para cualquier sitio web, pero más aún si se trata de un tutorial con fines didácticos, con una fuerte carga de exposición de materia teórica. Hay varios elementos a tener en cuenta en esta cuestión. El ancho del texto, o longitud de línea, es la decisión más importante. Existen tres opciones básicas: texto largo, medio o corto. Cuanto mayor sea el largo de los renglones, más texto cabrá en una sola pantalla, y por tanto, menor necesidad de *scroll* o hiperenlaces. Sin embargo, ocupar todo el espacio disponible con texto, para aprovechar al máximo el espacio, puede ser contraproducente. En general, se considera como opción más recomendable un ancho medio. En los textos largos el ojo debe hacer un recorrido muy largo para terminar una línea, lo cual provoca rechazo, por el contrario, en los textos con líneas cortas, el ojo debe de hacer saltos de línea con mucha frecuencia, lo cual acaba provocando fatiga.

Se considera que un texto es de ancho medio cuando tiene entre 60 y 80 caracteres por línea. Asumiendo un ancho de línea medio o corto, surge, además, la posibilidad de utilizar varias columnas, para aprovechar mejor el espacio, sin embargo, es una práctica desaconsejada ya que no suele facilitar la lectura. El interlineado y los márgenes son otros elementos a tener en cuenta, hay que elegir un valor lo suficientemente grande para facilitar la lectura, aún a costa de reducir la cantidad de información en pantalla, que puede obligar a utilizar un *scroll* o paginación.

La tipografía es el último elemento a tener en cuenta. Hay que considerar, básicamente, dos opciones de tipografía: serif y sans-serif. Se recomienda utilizar, en general, tipografía sans-serif para textos largos pues su lectura es mucho más cómoda. Existen, además, ciertas tipografías diseñadas especialmente para pantallas, como Verdana y Georgia. No obstante, no es descartable utilizar varios

tipos de letra para elementos de diferente clase, lo que ayuda a su diferenciación, por ejemplo, utilizar serif para títulos y encabezados, y para textos largos, como se ha mencionado, sans-serif [36].

3.1.4.3.7 Diseño de la retícula de la web

La distribución del contenido de la página web debe planificarse de acuerdo a la información que se va a mostrar en ella, por tanto, el primer paso es obvio, se debe hacer una clasificación, al menos de forma básica, de los elementos principales que la van a componer y la disposición esencial que deberán tener. Una vez obtenido un primer boceto, el objetivo es hacer una división de la pantalla, como un lienzo, a base de guías que formen rectángulos, sucesivamente más pequeños, respetando algún tipo de norma, como puede ser la proporción áurea, que es la más utilizada, de modo que finalmente, el tamaño y disposición del texto y de los márgenes entre diferentes paneles de contenidos formen una composición armoniosa.

Es conveniente colocar un dintel, o cabecera, en la parte superior de la web, en cual, se inserta, habitualmente, el logotipo de la empresa u organización de la web a la izquierda y un título o mensaje corto sobre la identidad de la página, a la derecha. Asimismo, es recomendable incluir un menú, que de acceso a las diferentes áreas del sitio web, a la izquierda, con color y diseño diferenciado del resto de elementos, pero en armonía [36].

3.1.4.3.8 Maquetación de HTML y CSS

Si bien los aspectos técnicos de la implementación del diseño web corresponden a la parte del capítulo referida a las tecnologías web, es necesario hacer unos breves comentarios sobre la integración de la definición del estilo mediante CSS con el contenido de la web. La idea básica es que toda la maquetación debe realizarse en base a etiquetas div, que son contenedores básicos que albergan agrupaciones de información, es decir, los rectángulos definidos previamente en la fase diseño a la hora de crear la cuadrícula. Estas etiquetas permiten establecer posicionamiento, tamaño y propiedades básicas como fondo, márgenes, bordes, para lograr el aspecto deseado de la página web. La maquetación en base a tablas, muy popular en tiempos pasados por su sencillez e inmediatez, es una manera obsoleta de implementar la maquetación de una web

por múltiples razones, como su capacidad limitada para crear layouts con suficiente libertad, el incumplimiento de la coherencia semántica de HTML y CSS, ya que el significado de las tablas es la presentación de conjuntos de datos, no el posicionamiento del contenido, las dificultades de mantenimiento del sitio web que posteriormente aparecerán, etc. Por tanto, esta segunda opción queda totalmente desechada y se resalta la importancia de seguir las buenas prácticas recomendadas, que se sustentan en la utilización creativa de etiquetas div.

Otra buena práctica es utilizar un convenio de notación CSS descriptivo y coherente, cuestión que en numerosas ocasiones se descuida y da lugar a hojas CSS muy difíciles de manejar cuando se quiere mantener o mejorar la página, o incluso, durante el propio desarrollo de la versión inicial. En general, se recomienda utilizar nombres descriptivos que atiendan a la semántica del documento, por ejemplo, utilizar como id o class de las etiquetas div expresiones tales como "índice", "cabecera", "cuerpo", "contenido_principal", "contenido_secundario", "ítem", etc, evitando de este modo expresiones tales como "botónGrisEstrecho", "barraAzulInferior", etc [36].

3.1.5 Análisis y seguimiento del sitio web: Google Analytics

La fase final que cierra el desarrollo del sitio web, tras el planteamiento de los objetivos, el diseño, su desarrollo e implementación y la puesta en marcha, es la etapa de seguimiento de los usuarios y la verificación de su funcionamiento.

Para este proyecto es útil disponer de una herramienta que aporte información relativa a las visitas que recibe el sitio web y que partes del mismo son las accedidas, por cuantos usuarios y en qué momento. Estos datos son muy valiosos para averiguar, de un modo sistemático y concreto, el uso que recibe el sitio web y sus pautas, es decir, la forma en que es utilizado, para de este modo, hacer un seguimiento continuado de éste y conocer su grado de difusión, y además, tener un punto de referencia si se plantean modificaciones o ampliaciones y adquirir experiencia que podrá ser útil en el futuro a la hora de desarrollar nuevos sitios webs de temática similar, como tutoriales con fines didácticos.

La herramienta ideal para obtener esta información es Google Analytics. Se trata de un servicio ofrecido por Google, de carácter gratuito y sencilla

implantación, que genera estadísticas con una amplia y variada lista de variables y opciones para realizar el seguimiento de un sitio web. Ofrece la posibilidad de realizar análisis complejos para elaborar estudios de marketing en profundidad, orientados a la empresa, sin embargo, para el propósito que se pretende conseguir en este proyecto, se tomaran en consideración sus funciones más elementales para obtener datos básicos sobre las visitas recibidas y el comportamiento de éstas.

La inclusión y puesta en marcha de este servicio en el proyecto es una tarea que requiere un tiempo y esfuerzo mínimo. Tan solo es necesario crear una cuenta en el mismo, que da acceso a una aplicación web en la que se puede configurar los datos de seguimiento del sitio, siendo su URL el dato fundamental, para luego posteriormente acceder a los informes generados tras ello. Adicionalmente, todas las páginas que se desee que sean monitorizadas, deben incluir un script para que el proceso se efectúe, el cual, es proporcionado por la propia herramienta. No requiere, por tanto, instalación y se puede empezar a utilizar de forma inmediata, la aplicación web además tiene una interfaz intuitiva que puede manejarse sin conocimientos previos.

Se pueden destacar, como datos mas significativos proporcionados por Google Analytics:

Visitas

Recuento del número total de visitas que ha recibido el sitio web, considerando éste el dominio definido en la cuenta Google Analytics como objetivo de seguimiento.

Páginas vistas

Número de páginas que han sido vistas por los visitantes del sitio web. Las visitas se refieren a los accesos a un determinado dominio, y por cada página alojada en ese dominio, se tiene un recuento del número de usuarios que la han abierto.

Promedio de páginas vistas

Relación entre el número de visitas y el número de páginas abiertas. Teniendo en cuenta el número total de páginas que componen el sitio web, puede conocerse, en promedio, que porcentaje del sitio web exploran los usuarios.

Porcentaje de rebote

Proporción de visitas que tan sólo abren una página y abandonan el sitio web inmediatamente. El significado de este dato no es trivial, incluye tanto a los usuarios que abren la primera página del sitio y la abandonan por que no les resulta útil, como los que llegan a través de un buscador para tomar un dato concreto; en el segundo caso, por tanto, no se trataría de un fracaso.

Tiempo en el sitio

Tiempo total, en promedio, que los usuarios permanecen navegando en el sitio web. Este dato es esencial para conocer verdaderamente si los usuarios hacen un uso útil, extrayendo información, del sitio web, pues un número de visitas elevado, en sí mismo, no puede considerarse una evidencia del éxito del sitio.

Para ello, habría que estimar un período de tiempo razonable en el que se puede leer y estudiar el tutorial y compararlo con la medida realizada por Google Analytics. No obstante, no sería un dato concluyente, ya que en este aspecto, se mezclan tanto la visitas que llegan con el objetivo de realizar el tutorial de manera completa, con las que solo buscan un dato concreto (al igual que sucedía con el porcentaje de rebote).

Usuarios del sitio web

Cómputo del número de visitas recibidas de nuevos usuarios. El concepto de "usuario nuevo" se refiere a visitas realizadas desde ordenadores en diferentes localizaciones, por lo que no se tiene garantía de que realmente se trata de personas que acceden por primera vez al sitio web.

Porcentaje de visitas nuevas

Visitas recibidas por el sitio web procedentes de nuevas localizaciones. Este dato se trata, por tanto, de la proporción entre el número de usuarios nuevos recibidos en un período de tiempo frente al total de usuarios diferentes que han llegado a la página desde su puesta en funcionamiento.

Los datos indicados pueden analizarse en diferentes escalas temporales, como días, meses y años. Además, pueden visualizarse en un gráfico basado en un eje temporal para facilitar su inspección.

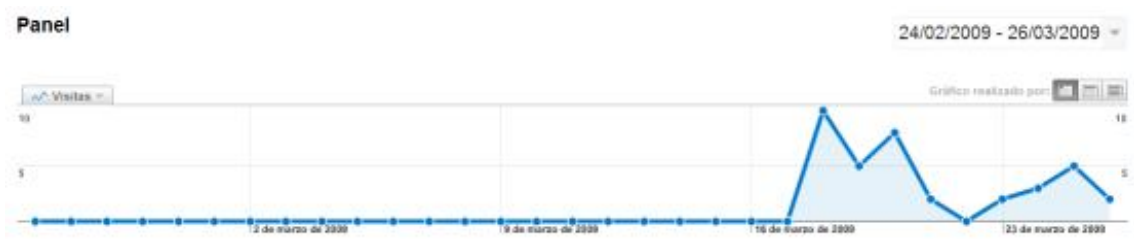


Figura 3-3 Gráfico Google Analytics del número de visitas recibidas en eje temporal

Google Analytics ofrece, además, datos de interés más avanzados sobre las visitas recibidas, como su procedencia geográfica, ubicación de red, y el entorno que el usuario utiliza para acceder al sitio web (navegador y sistema operativo).

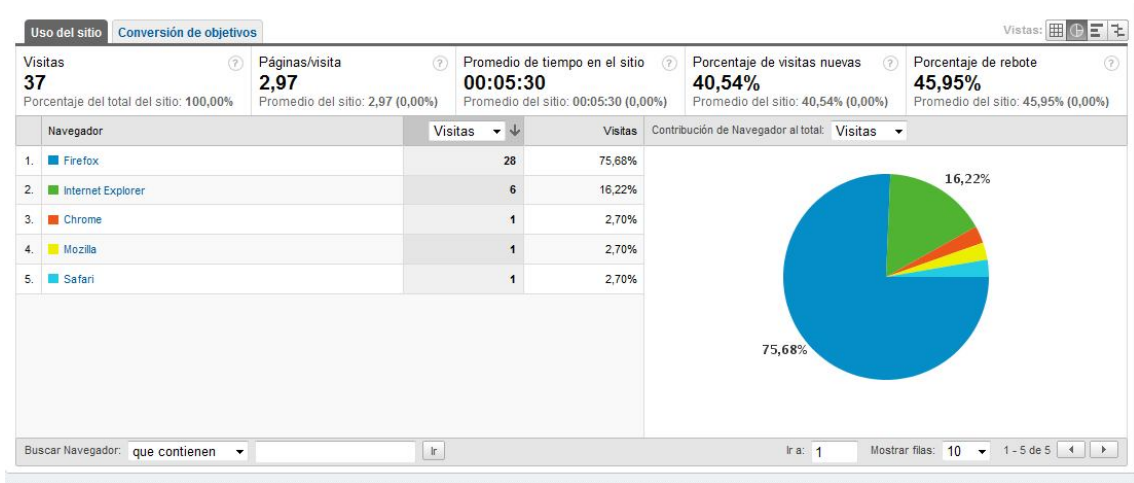


Figura 3-4 Gráfico Google Analytics de visitas en función del navegador web



Figura 3-5 Gráfico Google Analytics de la procedencia geográfica de las visitas

3.2 Lenguaje de desarrollo Java

Java es el lenguaje de programación sobre el cual se han desarrollado las aplicaciones incluidas en el tutorial, denominadas, técnicamente, “applets”.

Este lenguaje es una tecnología desarrollada por Sun Microsystems que tiene una serie de cualidades especiales que es conveniente considerar por tener una gran relevancia para el proyecto desarrollado.

Entre ellas, se puede destacar:

- Orientado a objetos
- Programación de alto nivel
- Multiplataforma
- Capacidad de desarrollo de aplicaciones para entorno web

Los programas Java se ejecutan a través de una máquina virtual que el usuario debe tener instalada y que ejerce de traductor entre el código creado por el usuario (binarios tras compilar) y las rutinas propias del sistema operativo. De este modo, para cada sistema operativo con relevancia en la actualidad, Sun ha desarrollado una máquina virtual específica y cuando se desarrolla una aplicación Java, tan sólo es necesario desarrollar una implementación, la máquina virtual se encargará de ejecutarlo de manera apropiada en cada plataforma [26].

3.2.1 Características

3.2.1.1 Orientación a objetos

La programación orientada a objetos es uno de los grandes paradigmas actualmente vigentes en el mundo de la programación, siendo, de hecho, el más utilizado. La programación estructurada es la otra visión existente la cual es más tradicional pero está siendo sustituida por la orientada a objetos.

Entre los lenguajes de mayor importancia orientados a objetos pueden destacarse C++, VB.NET y el comentado, Java.

Este tipo de programación se popularizó a partir de los años noventa por las grandes posibilidades que ofrecía a la hora de estructurar y planificar el código de aplicaciones, pues solventaba problemas de consistencia, coherencia, reutilización de código, etc.

La programación orientada a objetos aporta una abstracción que permite diseñar las aplicaciones en torno a entidades que facilitan el diseño del código, su reutilización y la comunicación entre diferentes módulos.

Las entidades básicas a considerar son:

Clase

Definición de un tipo de objeto con una serie de propiedades y métodos específicos.

Objeto

Entidad que se crea como una instancia de una clase y se comporta de acuerdo a la definición especificada en dicha clase que lo ha generado. De este modo, tendrá una serie de propiedades (variables) con cierto estado y una serie de métodos que realizarán tareas de diverso tipo, que pueden modificar las variables del objeto que las contiene o de otros objetos.

Método

Entidad que ejecuta código que realiza una tarea determinada. Los métodos pueden recibir variables por parámetros como datos de entrada y devolver una salida calculada a partir de dichas variables.

Evento

Sucesos que se producen durante la ejecución del programa, en un instante cualquiera no previsible, que pueden ser provocados por el usuario (por ejemplo, pulsar un botón, etc) o por algún agente externo (por ejemplo, llegada de un paquete a través de la red). Son notificados inmediatamente pudiendo el programa interrumpir la ejecución y tratarlos, para producir una respuesta adecuada o bien ignorarlos si no son relevantes.

Los programas, por tanto, se diseñan creando clases que definen un determinado concepto, pudiéndose referir a cosas reales (un usuario, si por ejemplo, se construye una aplicación que maneja datos de usuarios) o conceptos abstractos. En tiempo de ejecución, se crean instancias de estas clases que son los objetos, los cuales, se manejan para aplicar la funcionalidad de la aplicación, guardando el estado de todas sus variables que serán modificadas por sus métodos.

Esto implica que los programas pueden construirse de manera muy compacta, dividiéndolos en pequeñas unidades que se comunican entre sí con una serie de variables perfectamente protegidas.

Los programas se ejecutan a partir de un main que es un método que tendrá que estar contenido en alguna de las clases definidas. En él, se instancian objetos y se aplican métodos. Existe un método especial, el constructor, que se ejecuta, por defecto, en cada instanciación de un objeto y que puede servir como configuración inicial de las variables propias de esa clase.

La herencia es otro concepto esencial de la programación orientada a objetos, pues una herramienta poderosa para realizar reutilización de código y ampliar diseños.

Consiste en crear clases a partir de otras para aumentar su complejidad, pudiendo crear subtipos con unas características determinadas. Las clases “madre” contienen variables y métodos comunes de un determinado concepto y a partir de ellas se crean clases “hijas” que añaden funcionalidades para ramificaciones de dicho concepto utilizando la palabra reservada “extend” en su declaración, la cual permite realizar la ampliación sin necesidad de codificar de nuevo todo el código creado anteriormente en la clase madre, si no que se “hereda” [26].

3.2.1.2 Programación de alto nivel

El lenguaje Java ha sido diseñado desde un punto de vista de alto nivel, es decir, se trabaja en una capa elevada respecto al sistema operativo, lo que quiere decir que muchas de las tareas que tradicionalmente tienen que realizarse en comunicación directa con el sistema operativo, en Java, son encapsuladas por el motor de ejecución (la máquina virtual) de manera que para el desarrollador se simplifica notablemente la tarea de implementar las diferentes funcionalidades del programa, pues delega ciertas responsabilidades en la comentada máquina virtual.

Un ejemplo importante a considerar en el que se pueden ver las implicaciones que este método de trabajo supone es la gestión de memoria. En C, por utilizar como ejemplo un lenguaje de programación tradicional, la gestión la debe realizar personalmente el programador mediante reserva directa de trozos de memoria que deben ser liberados cuando ya no son necesarios. Si se producen inconsistencias en esta tarea, existe el riesgo de que se produzcan fugas de memoria que son trozos de memoria que quedan reservados sin posibilidad de ser liberados. El rendimiento y uso de memoria del programa dependerá de la habilidad del programador para gestionar correctamente este proceso.

En Java, en cambio, la realización de este proceso es transparente para el desarrollador. Éste puede crear libremente estructuras de almacenamiento tales como variables, objetos, arrays de objetos, listas, etc sin preocuparse de que sucede con ellas cuando se abandonan, pues la máquina virtual posee el llamado “recolector de basura” que es un proceso encargado de gestionar la memoria liberando la que corresponda cuando sea necesario.

Ésta y otras características de Java lo convierten en un lenguaje de programación sencillo, que permite desarrollar aplicaciones complejas con gran rapidez respecto a los lenguajes tradicionales como C, lo cual es una ventaja a considerar a la hora de desarrollar las aplicaciones planteadas en el proyecto, sin embargo, precisamente, su sencillez impone limitaciones que implican desventajas que serán comentadas más adelante [26].

3.2.1.3 Multiplataforma

Java cuenta como uno de sus principales puntos fuertes su capacidad para crear aplicaciones multiplataforma, es decir, que con una misma implementación, pueden correr en sistemas operativos o plataformas diferentes. Esta cualidad ha sido el motivo por el cual se ha popularizado de manera espectacular a partir de los años noventa. De hecho, más allá del simple hecho de poder crear un programa que puede funcionar tanto en Windows como Linux y Macintosh, la idea original es crear un lenguaje que pueda correr embebido en dispositivos muy heterogéneos y diferentes, más allá del tradicional PC, tales como móviles, electrodomésticos, etc. [25].

3.2.1.4 Capacidad de desarrollo de aplicaciones para entorno web

La orientación de Java al mundo web es otra de sus principales características. Por un lado, en lado del servidor, Java ofrece poderosas librerías y utilidades, tales como J2EE, que permiten desarrollar aplicaciones complejas que funcionan gestionando la interacción del usuario con páginas web dinámicas que implementan funcionalidades avanzadas que hacen uso de bases de datos. No obstante, esta parte de Java no es interesante para el proyecto, pues no se ha definido como necesidad disponer de una base de datos o registrar usuarios.

En lado del cliente, Java ofrece la posibilidad, como se ha explicado anteriormente, de incrustar aplicaciones complejas con funcionalidades mucho más avanzadas que las que puede ofrecer el HTML y tecnologías relacionadas (CSS, Javascript). Es ésta la opción que precisamente es más interesante para el proyecto.

Dichas aplicaciones son las applets, que son programas Java que pueden ejecutarse directamente embebidos en una página web, sin necesidad de ser

instaladas. En el fondo, la applets ahondan en la filosofía multiplataforma del lenguaje Java, pues el navegador web se puede considerar un dispositivo como otro cualquiera, con sus características especiales. Esto hace que no obstante, las applets impongan limitaciones para cumplir ciertos requisitos de seguridad, tales como acceder a ficheros de disco duro, descarga de código malicioso, etc.

El objetivo de las applets Java, en resumen, es dar la posibilidad de incluir en páginas web pequeñas aplicaciones, que son demasiado complejas para poder crearlas simplemente con HTML, pero que por su naturaleza, es deseable que puedan ejecutarse de forma inmediata, sin necesidad de descargarlas e instalarlas en el ordenador, ya que no tienen la suficiente entidad para ser consideradas un programa.

Las applets, no obstante, se encuentran en una situación incierta en el ámbito de las tecnologías web, pues existen otras posibilidades que ciertamente, ofrecen nuevas posibilidades desde puntos de vista originales. Por un lado, las aplicaciones Flash permiten crear presentaciones espectaculares con vistosos efectos especiales y por otro, el mundo AJAX, que consiste en la creación de aplicaciones semejantes a las tradicionales de escritorio utilizando simplemente XHTML y Javascript, está ganando una gran relevancia.

Sin embargo, ninguna de estas novedosas tecnologías puede todavía ofrecer las posibilidades que las applets de Java brindan. Las aplicaciones de simulación de procesos de tratamiento digital de audio que se han planteado para este proyecto exigen unos requerimientos tales como acceder al flujo de audio a nivel de muestra, manipular conjuntos de datos grandes como son los ficheros de audio y finalmente, insertarlos en el sistema de sonido del PC para reproducirlos por los altavoces, que de ninguna manera podrían ser implementados con otra tecnología que no sean las applets [26] [27].

3.2.1.5 Desventajas

Se han enumerado toda una serie de ventajas que hacen que Java y en especial, el módulo para crear applets, sea la plataforma mas indicada para desarrollar las aplicaciones que es necesario implementar en este proyecto, no obstante, Java tiene una serie de inconvenientes que hacen que no sea el lenguaje

de programación más habitual para desarrollar aplicaciones complejas para PC como lo son C++ o .NET.

Java es un lenguaje de programación lento y pesado. En cuanto a la velocidad, en Java un algoritmo cualquiera puede tardar unas veinte veces más en ejecutarse que en otros lenguajes como C. Respecto al uso de memoria, las aplicaciones Java ocupan mucho más, haciendo una gestión mucho menos eficiente de ella. Por otro lado, existen algunos condicionantes relacionados con el usuario que conviene considerar; para ejecutar una aplicación Java es necesario tener instalada, obviamente, una máquina virtual, lo cual puede suponer una molestia para el usuario si todavía no la tiene, aunque no obstante, es un problema menor, ya que los sistemas operativos mas populares como Windows o Macintosh traen una de serie.

Hay que notar que Java tiene unas características especiales que aportan ventajas pero que también tienen un reverso negativo por lo que el uso de este lenguaje dependerá de los requerimientos del proyecto a desarrollar. Por ejemplo, la gestión de memoria automática que realiza el "recolector de basura" es una ventaja por un lado, ya que evita el uso de punteros y memoria dinámica, simplificando enormemente la programación, pero por otro, quita control al desarrollador, resultando que en numerosas ocasiones se realiza una gestión ineficaz de ella, que se podría afinar en otros lenguajes como C.

3.2.1.6 Conclusiones y notas

Para el propósito de este proyecto, la implementación en Java está plenamente justificada, al margen de modas o tendencias volubles, de hecho, es la única plataforma que permite, a día de hoy, insertar una aplicación del tipo que se ha planteado, la cual realiza tareas complejas como aplicar algoritmos sobre streams de audio. Flash y AJAX son tecnologías web demasiado simples para este objetivo, mas bien, están orientadas a manejar presentaciones o crear aplicaciones de escritorio y por otro lado, la creación de una aplicación tradicional en C++ por ejemplo, rompería con la filosofía que se ha determinado seguir en este proyecto, que es la integración de las aplicaciones con la presentación de la teoría en un sitio web.

Se pueden destacar como componentes de Java que tienen relevancia para este proyecto algunos como Swing, dedicado a la creación de la interfaz gráfica, JavaSound, cuya función es la manipulación de flujos de audio así como otros de tipo auxiliar como Math, para realizar operaciones matemáticas.

3.2.2 Módulo de sonido JavaSound

JavaSound es un paquete de Java que contiene un conjunto de clases creadas para posibilitar el desarrollo de aplicaciones que trabajan con flujos de audio.

Se divide en dos ámbitos fundamentales. Por un lado, `java.sound.sampled`, dedicado al manejo de flujos de audio formados por muestras, como puede ser un archivo WAV, y por otro lado, `java.sound.midi`, dedicado a la manipulación de sonido MIDI. Para este proyecto, es el primero el que tiene interés.

La funcionalidad básica la ofrece aportando métodos que pueden realizar operaciones referidas a la entrada de audio, como son la lectura de un fichero en disco o la captura de una entrada de línea de la tarjeta de sonido, así como operaciones referidas a la salida, como la escritura en un archivo (WAV, por ejemplo) o la emisión de un flujo de audio por una salida, por ejemplo, los altavoces del PC. Asimismo, da la posibilidad de aplicar procesos intermedios, como mezclar flujos de audio de diferente origen para combinarlos y emitirlos por una salida o realizar procesamiento de audio elemental para regular la ganancia de los diferentes flujos o aplicar efectos como reverberación.

En definitiva, JavaSound es una interfaz de acceso a la tarjeta de sonido del PC para realizar las operaciones sobre ella que un desarrollador de software dedicado al audio puede necesitar.

Para facilitar esta tarea, pues los procesos tales como reproducción o captura de sonido son complejos, el paquete ha sido creado empleando la metáfora de la mesa mezclas, con flujos de entrada, de salida, y canales de mezcla.

La clase principal, a partir de la cual se crean todos los elementos necesarios para construir aplicaciones de audio, es `AudioSystem`. Representa una interfaz con la tarjeta de sonido mediante la cual se pueden pedir recursos. `AudioSystem` posee

métodos estáticos que devuelven, por ejemplo, objetos de tipo `AudioInputStream` o `Line`.

Los objetos de tipo `AudioInputStream` representan una señal de audio que entra o sale del sistema, como por ejemplo, un fichero WAV, y asociado a ella, tienen que existir siempre los objetos `AudioFileFormat` y `AudioFormat`, los cuales contienen datos referentes a la codificación y características de la señal, tales como el formato (PCM, por ejemplo), número de canales que lo forman, frecuencia de muestreo, etc.

La lectura de un archivo de audio que se ha definido creando el mencionado `AudioInputStream`, a partir, por ejemplo, de la ruta de un fichero WAV, se realiza de forma secuencial, mediante el método `read`, el cual almacena en bloques sucesivos en un array de bytes las muestras que forman la señal de entrada. El array de bytes obtenido se puede manipular libremente dentro de la aplicación desarrollada para luego sacarlo por altavoces o guardarlo en un fichero. Es importante resaltar que el byte no es un tipo de variable que suela ser el más idóneo para introducirlo en un algoritmo. Muy frecuentemente, se querrá transformar a una variable con precisión `double` o `float`, por ejemplo, que oscile en el fondo de escala del audio, por ejemplo, de 1 a -1. Para ello, `JavaSound` no aporta funcionalidad alguna, es una tarea que el desarrollador deberá implementar personalmente o tendrá que recurrir a librerías.

En cuanto a la entrada y salida de flujos de audio por la tarjeta de sonido, los elementos principales a considerar son las líneas y los mezcladores.

El fundamental es la línea (`Line`), que representa un canal de entrada/salida de la tarjeta de sonido. Es semejante al canal de una mesa de mezclas, que se puede utilizar como entrada de una fuente de sonido o bien, como mezclador al cual se mapeen diferentes canales de entrada para elaborar una mezcla. Asociado a una línea (`Line`) se encuentra siempre un objeto de tipo `Control`, que representa los controles que se le pueden aplicar a ella, tales como algunos de los mencionados anteriormente, como regulación de ganancia, reverberación, etc.

Dado que la funcionalidad de `Line` es de propósito muy general, existen una serie de clases que extienden de ella y que tratan casos concretos. Por ejemplo,

Mixer, representa un punto de mezcla de diferentes flujos de audio. SourceDataLine, por otro lado, representa un punto en cual introducir muestras de audio que han sido procesadas por la aplicación y que se desean sacar para ser reproducidas. Posee el método write que actúa escribiendo en un buffer bloques de muestras que son enviadas a la tarjeta de sonido para emitirlas por los altavoces. Son estas clases las que se utilizan en la práctica para llevar a cabo las operaciones sobre flujos de audio [27] [28] [29].

3.2.3 Librería de gráficas JFreeChart

3.2.3.1 Introducción

JFreeChart es una librería Java para la presentación de gráficas de alta calidad.

Se trata de un proyecto iniciado en el año 2000 por David Gilbert y que es mejorado y ampliado continuamente. Se ha convertido en una de las librerías para la creación de gráficas de mayor éxito y difusión en el mundo Java, por la calidad de los resultados y su versatilidad. Es un producto libre, con licencia LGPL (Lesser General Public License), por lo que se puede utilizar, modificar y distribuir con libertad.

Esta librería ofrece la posibilidad de representar series de datos en múltiples tipos de gráficas: de líneas, de barras, gráficos de tarta, gráficos estadísticos. Cada una de ellas ofrece una gran variedad de opciones y parámetros configurables, como la disposición, el tamaño, la presentación, combinación de diferentes series de datos, etc.



Figura 3-6 Ejemplos de gráficas JFreeChart

Si bien tiene numerosas herramientas para la creación de gráficos orientados a estadística y economía, es también una librería adecuada para la representación de señales [30] [31].

3.2.3.2 Integración en las aplicaciones del tutorial

En el caso concreto de la representación de señales, y teniendo en cuenta los requerimientos del proyecto, la herramienta de interés principal de JFreeChart para las aplicaciones es la clase XYLineChart, cuya utilidad es pintar gráficos en dos dimensiones de series de pares de datos, representados por puntos unidos por líneas. Este componente es el indicado para pintar la forma de onda de la señal de audio en el tiempo (formada por pares de datos de tiempo y nivel de señal) y de la densidad espectral de potencia (formada por pares de datos de frecuencia y potencia). Entre sus cualidades principales es destacable la posibilidad de configurar el rango de los ejes, la presentación de las etiquetas de dichos ejes, los títulos y leyendas, así como el aspecto general de los elementos representados y su disposición. Es posible, además, añadir utilidades de gran interés para el manejo de las applets, como el zoom sobre la gráfica o la posibilidad de guardar la misma como una imagen PNG.

El uso de esta librería es sencillo y permite crear las gráficas de una manera fácil y práctica. Ofrece una clase principal "ChartFactory" que genera gráficas del tipo especificado, recibiendo como argumentos objetos de la clase XYSeries, que almacenan las muestras, así como algunos parámetros de configuración. Devuelve un objeto tipo JPanel que puede ser insertado en los elementos gráficos de la interfaz visual de las applets, pudiendo ser luego modificados si es necesario [30] [31].

3.2.3.3 Estudio del rendimiento

El rendimiento que ofrece esta librería cumple con suficiencia los requisitos de la aplicación a desarrollar. Con un equipo informático de gama media, permite generar las gráficas en tiempos razonables, de modo que su integración en la aplicación es óptima. No obstante, hay que tener presente que dada la complejidad de los objetos que genera, si la cantidad de muestras que forman las series de datos a representar es muy elevada, pueden producirse retardos.

Se pueden citar algunos ejemplos prácticos que se han realizado antes de confirmar la elección de esta librería como sistema para generar gráficos.

Un array de 256 muestras provenientes del cálculo de la FFT sobre una ventana de la señal en un instante concreto, es representado de forma inmediata (menos de un segundo), por lo que el retardo es inapreciable. La implementación, por tanto, de un scroll para recorrer la señal en el tiempo mostrando la evolución de la FFT en el tiempo es viable.

La representación de la señal en el tiempo es, en cambio, más problemática. El número de muestras que cabría requerir a la librería representar alcanza unos valores que hacen que el coste computacional sea considerable.

Para una señal de audio típica, de calidad CD, muestreada a 44.1 KHz, es necesario pintar 44100 muestras para representar un segundo de señal en un panel gráfico, considerando que es una señal mono o al menos, interesa sólo uno de los dos canales si es estéreo. Pruebas realizadas sobre equipos informáticos de diferente potencia han mostrado que pueden llegarse a necesitar varios segundos de tiempo para realizar el proceso. Esto supone un grave inconveniente que podría desaconsejar el uso de JFreeChart.

Se ha logrado obtener una solución de compromiso que hace viable la utilización de la librería JFreeChart para este propósito. Para señales muestreadas a 11.025 Hz, la representación de medio segundo de señal significa introducir 5512 muestras. En ordenadores con una potencia media se ha comprobado que el tiempo de proceso de esta tarea es viable, y que alcanza una cota que hace posible un uso cómodo de la aplicación, apenas un retardo de medio segundo, aproximadamente. Hay que tener en cuenta que el propósito principal de la aplicación es utilizar señales de voz mono, para las cuales, la utilización de 11025 Hz de frecuencia de muestreo es suficiente. No obstante, si se quisiese utilizar una señal de calidad CD de 44.1 KHz, no sería problemático, pues sólo supone un tiempo de proceso añadido proporcional al número de muestras, que es perfectamente asumible para el usuario si la utilización de una señal de esas características es importante.

Dado que las señales que se ofrecerán al usuario serán mayores que medio segundo, será necesario una ofrecer una solución en la interfaz gráfica para permitir la navegación por la señal. Un elemento tipo scroll o botones de avance y retroceso solucionan este requerimiento de forma sencilla, ofreciendo al usuario un sistema de navegación claro y eficaz.

Como desventaja más importante a tener en cuenta, hay que señalar que no ofrece ninguna utilidad dedicada específicamente a la creación de gráficos dinámicos que varíen en el tiempo, con lo cual, si se requiere crear una gráfica en la cual los datos se van actualizando de forma periódica (como puede ser la transformada de Fourier de ventanas de señal sucesivas en el tiempo), la implementación tendrá que ser realizada por el propio desarrollador. Esta característica, para la aplicación en desarrollo, no es un requerimiento básico, por lo que ha quedado descartada y queda pendiente de implementación para una versión posterior.

En conclusión, JFreeChart es una librería idónea para resolver las necesidades de representación gráfica de señales de la aplicación. Ofrece la creación de gráficas con un aspecto visual agradable y atractivo, con las suficientes opciones de configuración para presentarlas del modo que interesa. Su integración en las clases que componen las applets es sencilla y ahorra un tiempo de desarrollo de código notable, lo que la convierte en la mejor opción, en contraste con otras posibilidades, como el desarrollo de una librería gráfica propia (que excede notablemente los requisitos de tiempo y trabajo especificados). Las cuestiones de rendimiento han sido estudiadas en detalle antes del desarrollo de los módulos del proyecto concernientes a las gráficas, para asegurarse de que se obtendrán resultados óptimos antes de integrarlas en el mismo. Teniendo en cuenta ciertas restricciones de coste computacional, se ha logrado encontrar una solución viable y perfectamente válida. Hay que tener en cuenta que trabajar con señales supone manejar grandes series de datos y que en un lenguaje como Java, aparecen dificultades inevitables.

3.2.3.4 Uso de las gráficas

Respecto al uso que se le va a dar a las gráficas en las applets, pensando en la experiencia del usuario, se han considerado una serie de situaciones de uso en las JFreeChart aporta una solución óptima.

Las gráficas en tiempo aportan una visión gráfica general de la forma de onda de la señal, para hacerse una idea del rango dinámico de ésta y su evolución. La información que aportan es útil para conocer el nivel de señal en cada tramo y relacionarlo con el comportamiento de las técnicas estudiadas.

El uso, por tanto, de capturas estáticas de medio segundo de señal, fácilmente navegables mediante botones de avance y retroceso, es adecuado.

Una tarea fundamental en el uso de las applets es la inspección del resultado de la cuantificación en señales de bajo nivel y su relación con el *dither*. Es importante que el usuario visualice la acción del *dither* sobre señales, por ejemplo sinusoidales, de un nivel muy bajo, del orden de un escalón de cuantificación, para que aprecie la basculación que introduce el *dither*, lo cual, elimina la distorsión provocada por la aparición de pulsos cuadrados. Para esto, la herramienta de zoom de JFreeChart es muy útil y se ha tenido en cuenta a la hora de la elección de esta librería.

Asimismo, en otras situaciones, como la inspección del comportamiento de las técnicas estudiadas en zonas de silencio o casi silencio de señales audio comunes, la herramienta de zoom es también una gran ayuda.

En cuanto a la representación de la densidad espectral de potencia de una ventana de señal, es importante que ésta se realice en un tiempo muy corto, casi instantáneo, para que el usuario pueda observar la evolución de la misma a lo largo del tiempo de forma rápida, tanto de forma manual (con un scroll), seleccionando los instantes concretos que interesan, como de forma automática, a la vez que se reproduce al audio. Esta segunda posibilidad es opcional, y queda para futuras mejoras de la aplicación, teniendo en cuenta que su realización es viable. Para la representación de la DEP de una ventana de 256 muestras de señal, el resultado es inmediato, por lo que se cumplen los requisitos establecidos.

La herramienta de zoom también puede ser eventualmente útil en el caso de las gráficas de densidad espectral de potencia. Con ella, será posible observar con facilidad los armónicos que aparecen al cuantificar señales sinusoidales, en múltiplos de la frecuencia del tono, aunque estos sean muy pequeños si la cuantificación no se realiza con un número excesivamente bajo de bits.

IV

DISEÑO E IMPLEMENTACIÓN DEL TUTORIAL

4 Diseño e implementación del tutorial

4.1 Diseño web del tutorial

El diseño de la web del Tutorial de Técnicas de Digitalización de Audio se ha realizado basándose en los conceptos básicos de un buen diseño web, presentados anteriormente, con el objetivo de crear un curso online y fuente de referencia, que pueda ser utilizado como fuente de consulta de los conceptos teóricos de la materia y como lugar para realizar experimentación práctica sobre los sistemas y algoritmos tratados.

La integración de ambas vertientes, la exposición de la teoría y la realización de ejercicios prácticos es la clave del proyecto. Se pretenden combinar de un modo natural y coherente, intercalándose para que el alumno adquiera los conocimientos de forma progresiva, con la motivación extra que supone comprobar de forma inmediata, con muy poco esfuerzo, el funcionamiento real de las demostraciones matemáticas desarrolladas de forma teórica. En suma, se trata de aprovechar las ventajas del *e-learning* vistas al plantear este proyecto. Se persigue que el hecho de pasar de estudiar teoría a practicar con ejercicios sea muy natural, que el cambio se perciba lo menos posible.

Ante este reto surge un dilema de gran importancia a la hora de plantear la estructura del tutorial. Hay que decidir entre diseñar una navegación fuertemente guiada, de un modo en el que el usuario siga una serie de pasos bien definidos y marcados, sin posibilidad de avanzar, retroceder o saltar a cualquier punto de forma libre, y la contraria, una navegación en la que se presente todo el contenido desde un principio y sea el mismo usuario el que decida que partes del tutorial visitar y en qué orden.

El resultado deseable es obtener una solución que combine ambas opciones, pues no son excluyentes. Para conseguirlo, la mejor vía a seguir es diseñar una interfaz minimalista que con los elementos de navegación justos, sin recargar el sitio web de componentes gráficos innecesarios, facilite al usuario manejar el tutorial de forma secuencial, a modo de curso, pero también, que permita acceder de la forma más directa posible a cualquier parte del mismo.

El usuario entra en la web a través de una pantalla de bienvenida que muestra tres opciones: inicio del tutorial, ayuda y “acerca del tutorial”. Esta pantalla tiene como finalidad principal reforzar la imagen corporativa del tutorial, aunque no obstante, para dotarla de cierta utilidad, y que no consista sólo en una portada decorativa, en ella se incluye una breve descripción del contenido del tutorial, a modo de *abstract*, para invitar al visitante que acceda a él. Todas las opciones conducen a la vista principal del tutorial, la cual alberga una serie de pantallas que contienen los capítulos del mismo y algunos otros contenidos adicionales.

La vista principal del sitio web se ha realizado con un diseño basado en una solución simple, compuesta por un índice, siempre visible, junto a un área central que alberga el contenido de los capítulos del tutorial. De este modo, el usuario puede acceder directamente a cualquier capítulo del tutorial y nunca se perderá en el mismo, al tener siempre a mano el índice para saltar a cualquier parte en cualquier momento, estando resaltada la opción del capítulo en el que se encuentra situado en ese instante. Además, se incluyen al comienzo y final de cada capítulo enlaces al anterior y el siguiente, para complementar el índice y así facilitar la navegación, su presencia, por tanto, es redundante, pero enriquece la interfaz y ahonda en la metáfora de curso interactivo.

La vista principal contiene una cabecera, que al igual que el índice, tiene una posición fija y se mantiene a la vista en la parte superior cuando el usuario hace scroll para continuar leyendo un capítulo. En general, no se recomienda que las cabeceras ocupen mucho espacio o llamen demasiado la atención, si no tienen funcionalidades específicas como links. Por este motivo, la cabecera contiene tan sólo el logotipo de la universidad y el nombre del tutorial, al estilo de la mayoría de las páginas web actuales.

El área central de la vista principal contiene la información de cada capítulo del tutorial, a modo de página de un libro. El ancho del texto es de longitud media, con sus márgenes correspondientes, para facilitar la lectura, evitando que un renglón excesivamente largo provoque fatiga, y así, además, potenciar la metáfora de página de un libro. El largo es variable, y cuando se supera el espacio vertical, se puede recorrer todo el capítulo mediante un scroll. El índice, como se ha dicho,

tiene una posición fija, por lo que el capítulo se desplazará hacia arriba o hacia abajo pero el índice permanecerá anclado, siempre accesible para cambiar de capítulo.

Los capítulos que componen el tutorial se agrupan en cuatro categorías básicas de acuerdo al temario de las técnicas de digitalización de audio. De esta manera, se definen cuatro temas de estudio:

- La cuantificación
- Sobremuestreo
- Dither
- Noise-Shaping

En el índice, los capítulos se agrupan en cajas, una para cada tema de estudio, diferenciándose con claridad, de modo gráfico, las diferentes áreas temáticas del tutorial.

Se ha buscado que la longitud de los capítulos sea parecida, para crear una impresión de consistencia, de modo que el usuario se familiarice con el tutorial tras visitar unos pocos capítulos. Aún así, esta premisa no se ha podido cumplir totalmente, por las propias exigencias del material didáctico a exponer. En cualquier caso, se ha evitado crear capítulos excesivamente largos, con un scroll demasiado grande, para que no se abandone su lectura y además, facilitar la tarea a las personas que ya lo han leído y posteriormente lo revisitan para buscar un dato concreto, y por otro lado, también se ha evitado crear capítulos muy cortos, al estilo de diapositivas, para evitar la fragmentación del contenido. Asimismo, se ha procurado hacer una utilización adecuada de gráficos e imágenes, de modo que apoyen el aprendizaje de forma efectiva, evitando elementos decorativos que no aporten información, y en aquellas partes donde se requiere la exposición de demostraciones matemáticas, se han complementado con abundantes comentarios intercalados, que facilitan su comprensión.

Cada uno de los cuatro temas que componen el tutorial finaliza con un capítulo nombrado “Estudio práctico”. Este capítulo alberga la aplicación Java que implementa la parte práctica de cada tema junto con una “Guía de uso”, que consta de un diagrama del sistema y una serie de experimentos diseñados para estudiar

de forma práctica y de un modo sistemático los conceptos tratados en ese tema. Una vez se ha constatado que no es posible la inclusión de todo ese contenido en una sola pantalla, pues excede notablemente los límites del área central del tutorial, ha surgido la necesidad de crear un sistema que permita alternar uno y otro contenido de forma sencilla y rápida. Para ello, se optado por crear dos pantallas que se alternan en el mismo espacio.

En una de ellas, está incrustado el applet, y en la otra, una "Guía de uso", en la que aparece un gráfico con el sistema que implementa la aplicación y la serie de indicaciones y recomendaciones para el uso del applet, organizadas en experimentos. Ambas pantallas se intercambian de forma automática con un botón situado en la parte superior del capítulo. Cuando se pulsa el botón, se oculta la pantalla visible y se trae al frente la que estaba escondida. El motivo para elegir esta opción de diseño es maximizar el uso del espacio de la web. Tanto la colocación del applet en primer lugar y a continuación, toda la información relativa a la misma, como la posibilidad contraria, exponer en primer lugar las recomendaciones y finalizar la página con el applet, se han demostrado ineficientes desde el punto de vista de la usabilidad y el manejo, pues generaban un scroll demasiado amplio, que dificultaba el acceso inmediato a cualquiera de las partes y que acababa provocando fatiga. Inicialmente, aparece el applet y la información está oculta. Esto conlleva un riesgo evidente, pues se corre el peligro de que el usuario no preste atención a que puede acceder a información sobre el applet. Este problema es inevitable, pero se puede mitigar con el diseño de un botón bien visible y la inclusión de un texto inicial que anime al usuario a inspeccionar el gráfico del sistema.

Los últimos elementos que forman parte del tutorial son una serie de páginas que aportan información complementaria, se encuentran agrupadas bajo el epígrafe "Ayuda" y se componen de tres elementos. "Ayuda", en la cual se especifican una serie de requisitos y recomendaciones para el uso del tutorial y las applets; dicha información es de carácter exclusivamente técnico, para solventar posibles problemas informáticos, especialmente en la ejecución de las applets, las recomendaciones de carácter didáctico se hacen, como se ha comentado antes, en cada capítulo "Estudio Práctico". "Bibliografía y enlaces", en la cual, se aportan

todas las fuentes de referencia a partir de las cuales se ha extraído el contenido teórico del tutorial y que pueden ser utilizadas por el alumno para ampliar conocimientos; al respecto, hay que mencionar que para facilitar la navegación por el tutorial, cada vez que se cita una referencia bibliográfica, esta contiene un enlace a la página para localizar rápidamente la fuente; además, en los documentos bibliográficos disponibles en la Red, se aporta un enlace a su URL. Finalmente, se incluye una página "Acerca de..." en la que se aporta información relativa al autor, el desarrollo del proyecto, y los componentes del tutorial externos (librerías, código libre, etc).

4.1.1 Vistas de las áreas del tutorial

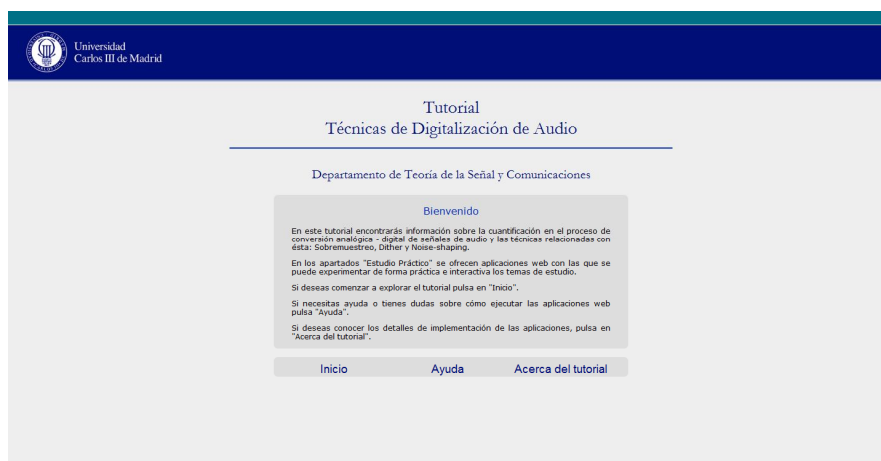


Figura 4-1 Menú de Inicio

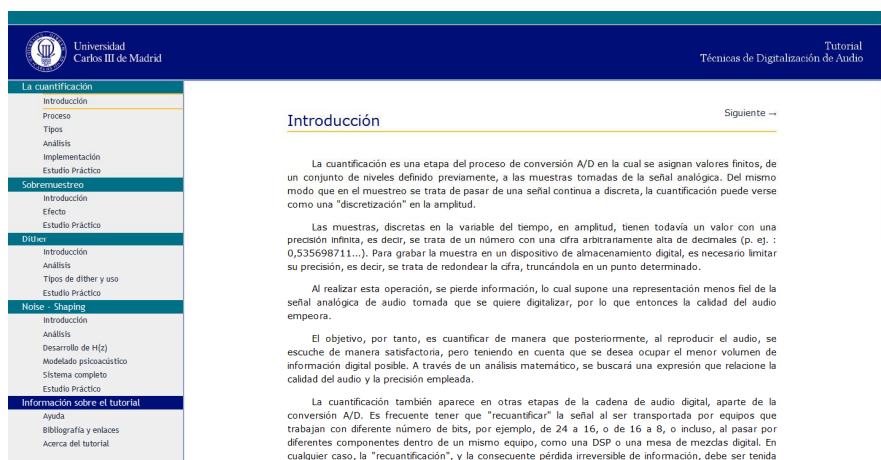


Figura 4-2 Vista principal

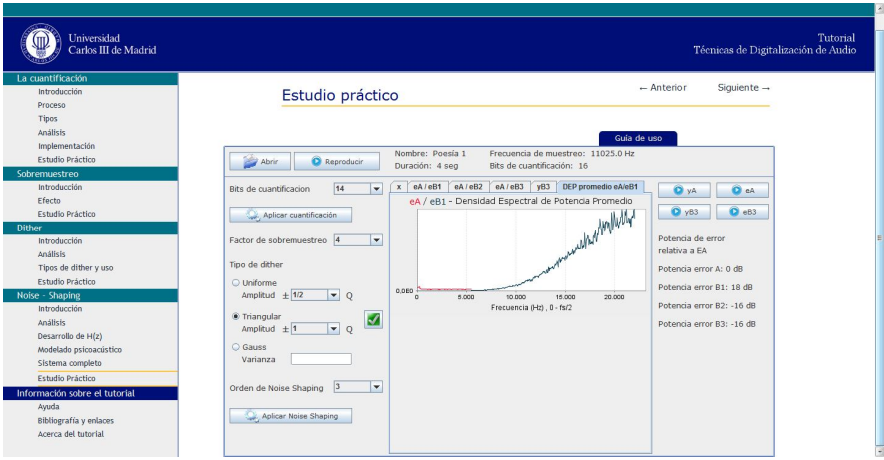


Figura 4-3 Estudio práctico (aplicación visible)

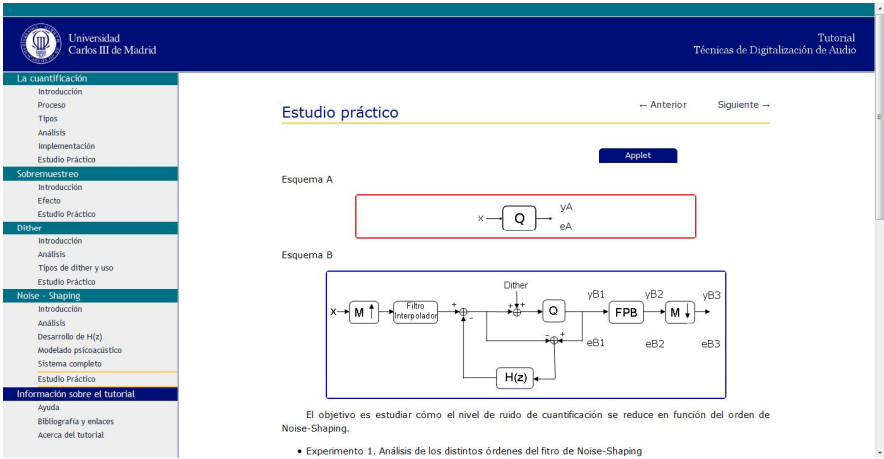


Figura 4-4 Estudio práctico (instrucciones visibles)

4.1.2 Esquema de organización del sitio web

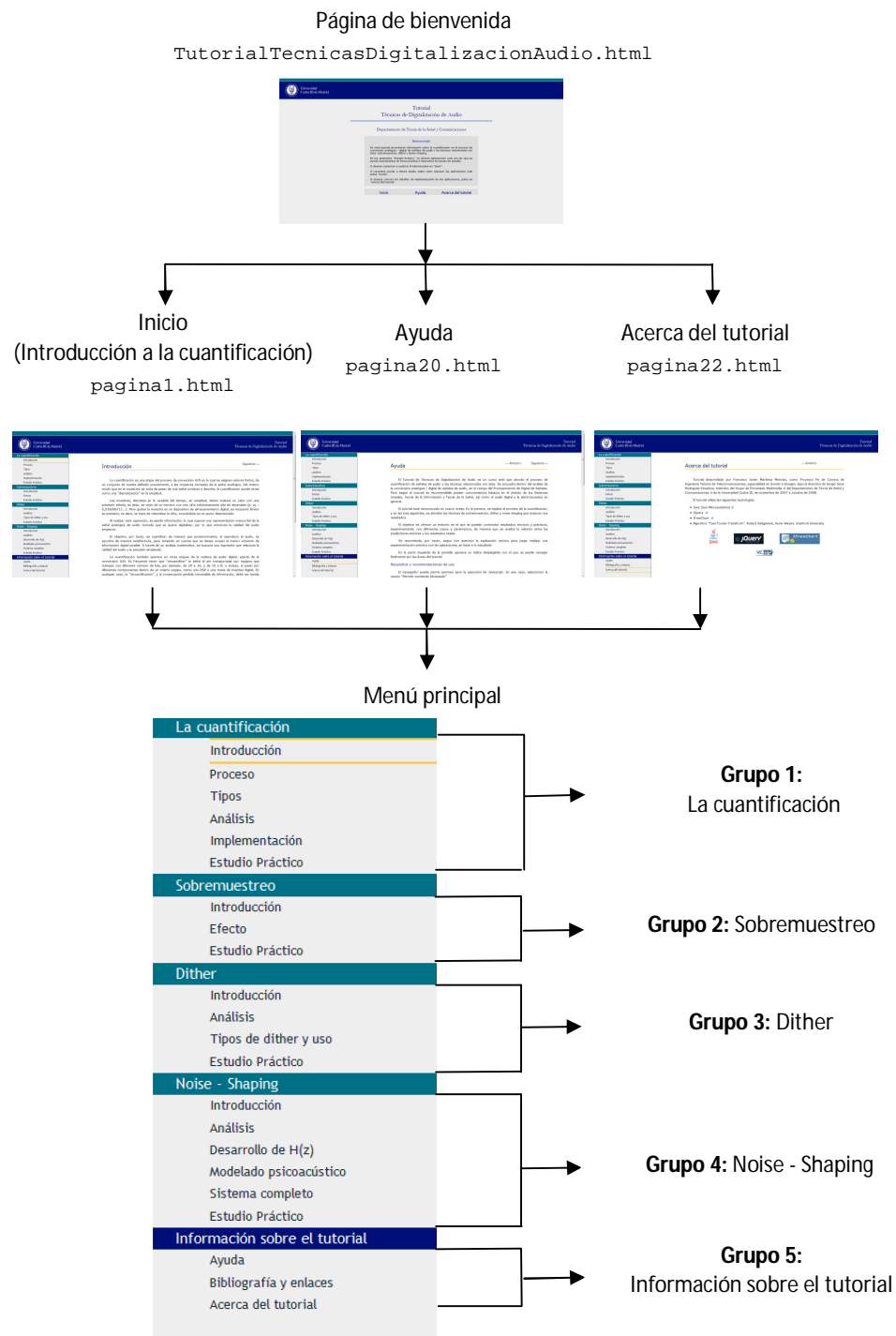
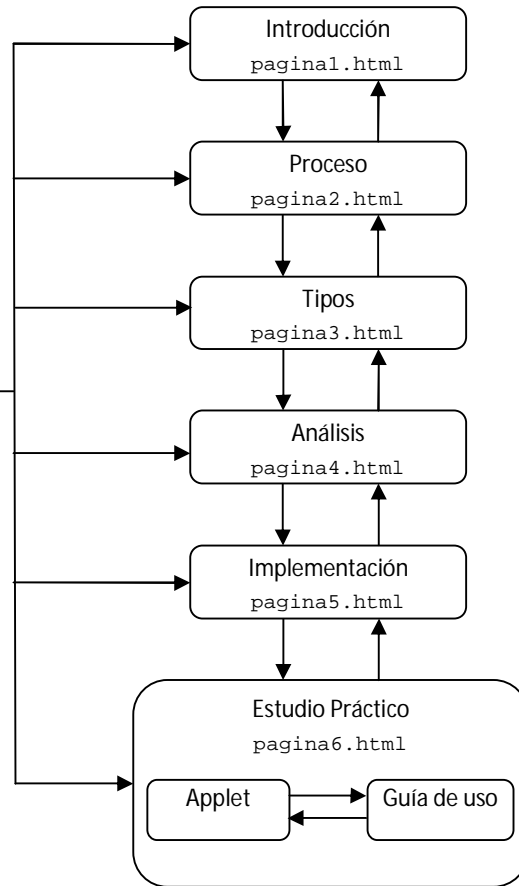


Figura 4-5 Esquema de organización del sitio web (I)

Grupo 1
La cuantificación

La cuantificación
Introducción
Proceso
Tipos
Análisis
Implementación
Estudio Práctico



Grupo 2
Sobremuestreo

Sobremuestreo
Introducción
Efecto
Estudio Práctico

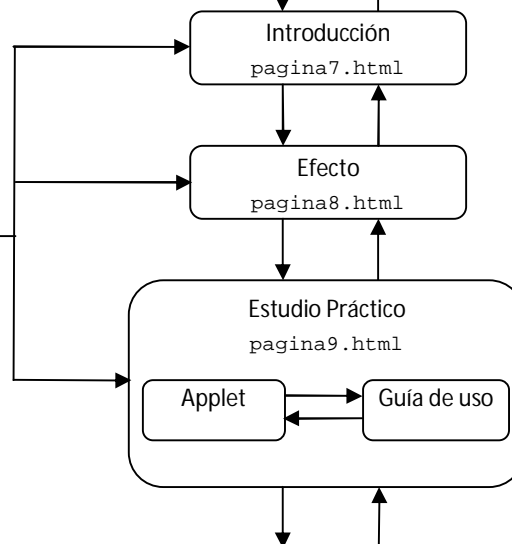
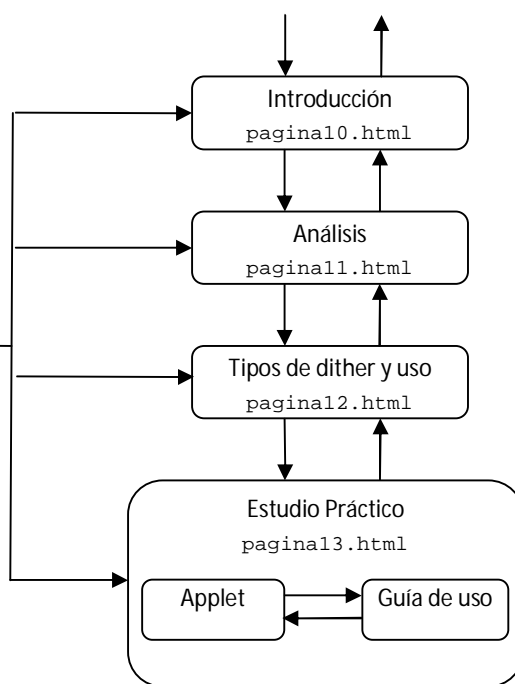


Figura 4-6 Esquema de organización del sitio web (II)

Grupo 3
Dither

Dither
Introducción
Análisis
Tipos de dither y uso
Estudio Práctico



Grupo 4
Noise - Shaping

Noise - Shaping
Introducción
Análisis
Desarrollo de H(z)
Modelado psicoacústico
Sistema completo
Estudio Práctico

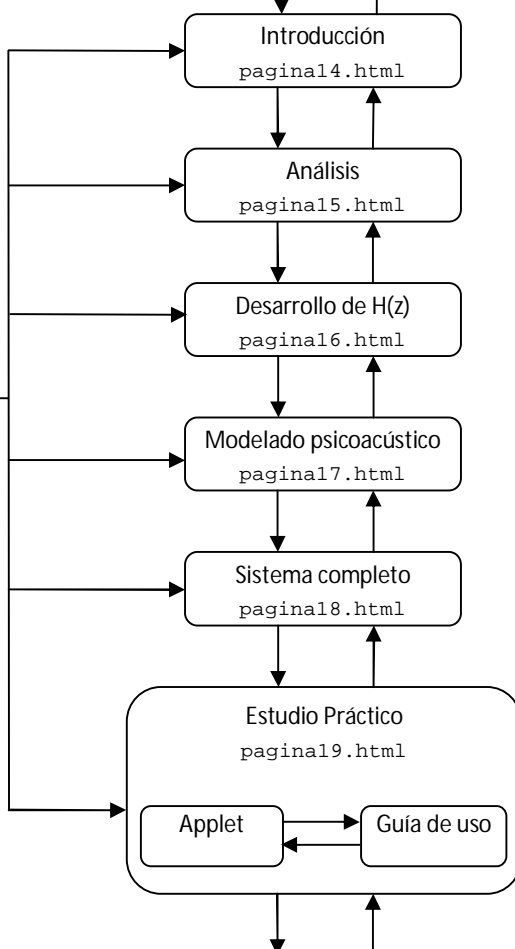


Figura 4-7 Esquema de organización del sitio web (III)

Grupo 5
Información sobre el tutorial

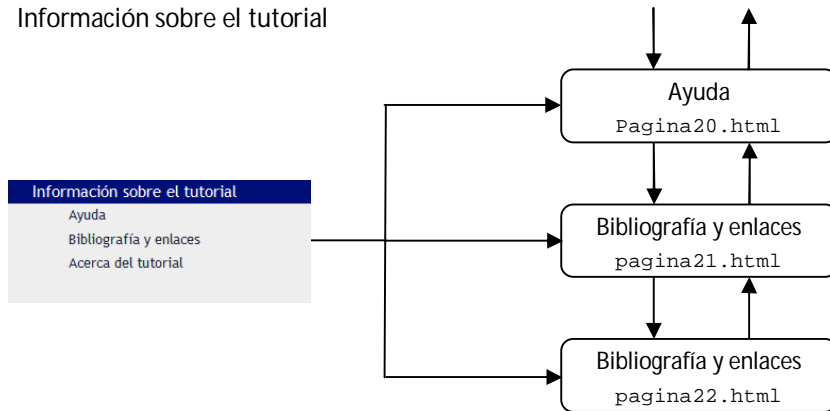


Figura 4-8 Esquema de organización del sitio web (IV)

A continuación se presenta una breve descripción de los capítulos que forman parte del tutorial.

La cuantificación

Introducción

Este capítulo inicial sirve para situar la operación de la cuantificación dentro del proceso de la conversión analógica – digital y explicar la esencia de su funcionamiento y el motivo por el que se aplica. Asimismo, se introduce el concepto de “ruido de cuantificación”.

Proceso

Se explica de un modo gráfico, de la manera más elemental posible, el proceso de la cuantificación a partir de una onda continua que pasa por un cuantificador de cuatro bits, observando los cambios que se producen en ella en diferentes pasos, en especial, la asignación de las muestras a los niveles más cercanos. El objetivo es ver del modo más simplificado posible el proceso, para entender con claridad, de un modo visual, que significa el concepto abstracto de “pasar de un valor de amplitud analógico a uno discreto”

Tipos

Una vez entendida la cuantificación en su procedimiento esencial, se presentan los diferentes tipos que existen, diferenciando básicamente la uniforme de la no uniforme y aplicación.

Análisis

Se desarrolla la demostración matemática que relaciona el número de bits empleado con la relación señal / ruido que se obtendrá. Este estudio es fundamental completarlo con una reflexión sobre el fenómeno de la distorsión y cómo la hipótesis de ruido de cuantificación modelado como ruido con fdp uniforme y blanco es sólo una aproximación. La comprobación de este fenómeno será uno de los fines principales del applet que completa este tema.

Estudio práctico

Se presenta el applet con un estudio práctico con varios experimentos.

El primer experimento está orientado a estudiar la reducción de la relación señal / ruido en señales de voz y música, apreciando subjetivamente la aparición del ruido de cuantificación mientras se reduce progresivamente el número de bits de cuantificación con los valores 12, 8 y 4.

El segundo experimento toma como señal de estudio un tono para observar la aparición de pulsos rectangulares en el tiempo y armónicos en el espectro (réplicas del tono en múltiplos del mismo) debidos a la cuantificación.

Implementación

Se describen, de forma breve, las ideas básicas de cómo se implementan, en la práctica, es decir, mediante componentes electrónicos y en DSP's, los sistemas cuantificadores. De este modo, se pretende que el alumno encuentre la conexión entre los conceptos teóricos y su realización práctica en componentes físicos, para así cerrar el círculo. Se aportan, además, algunos detalles sobre su aplicación en el audio, y en concreto, en la grabación profesional, que pueden despertar interés en un ingeniero de sonido, por lo que además, se presenta un esquema de una cadena de audio en la que está presente un conversor A/D en una de sus etapas.

Sobremuestreo

Introducción

Se presenta una explicación básica del funcionamiento del sobremuestreo, cómo se realiza y que elementos forman el sistema que lo implementa. Además, se incluye una explicación de la Teoría de Nyquist, como breve recordatorio de su papel en la etapa de muestreo, pues este concepto está íntimamente ligado a la técnica de sobremuestreo, y por tanto, para su correcta comprensión, es conveniente repasarlo.

Efecto

Se explica de forma gráfica de qué modo afecta al espectro del error de cuantificación la aplicación de la técnica del sobremuestreo, observándose de un modo muy visual e inmediato, como se reduce el error de cuantificación. Esta explicación se complementa con un segundo epígrafe en el que se indican otras ventajas colaterales en la digitalización de señales, que se pueden aprovechar al emplear esta técnica.

Estudio práctico

Se presenta el applet con un estudio práctico con varios experimentos.

El primer experimento tiene como objetivo constatar cómo se reduce en 3 dB el ruido de cuantificación cada vez que se duplica el factor de sobremuestreo. Para ello, se recomienda abrir una señal compleja (música) y emplear un número de bits de cuantificación alto (14). Con estas condiciones, se pretende aproximarse lo máximo posible al caso ideal de ruido blanco. El usuario deberá estudiar las tres etapas fundamentales por las que pasa la señal y observar las modificaciones que sufre: interpolación, en la que el ruido se extiende en el espectro (eB1); filtrado paso bajo, en la que se elimina el ruido que queda por encima del ancho de banda original (eB2); y diezmado, en la que la señal vuelve a tener la frecuencia de muestreo original pero posee un nivel menor de ruido de cuantificación (eB3).

El segundo experimento es similar al anterior, pero empleando unas condiciones en las que no se cumple la hipótesis de ruido blanco, utilizando un número muy bajo de bits de cuantificación (4), por lo que la disminución

progresiva en 3 dB del ruido no se cumple de manera tan exacta como en el primer experimento.

Se plantean al usuario, además, varias preguntas para que reflexione sobre la técnica estudiada: ¿Porqué las potencias e_{B2} y e_{B3} son iguales?, y ¿Cómo se escucharía la señal y_{B2} si se reprodujese a la frecuencia de muestreo original?

Dither

Introducción

Se explica el concepto de *dither* en esencia y cuál es la razón básica de su funcionamiento. Esta explicación se apoya haciendo especial hincapié en la comparación entre el efecto fisiológico que produce en el oído escuchar distorsión frente a escuchar ruido blanco, lo cual, explica la aparente contradicción que, inicialmente, pudiera parecer el hecho de añadir más ruido para lograr una reducción del mismo. La clave que hay que comprender es que no se trata de reducir el ruido, si no de transformarlo en otro tipo de ruido más agradable y menos incómodo.

Análisis

Una vez se ha comprendido el objetivo del *dither* y el resultado sobre la audición de una señal cuantificada, se muestra su efecto visualmente mediante una representación de la señal en tiempo y frecuencia, de forma esquemática y simplificada, es decir, con una señal idealizada y no una señal real, para comprender como afecta a las características de la señal. Para ello, se toma como referencia una señal consistente en una onda sinusoidal, pues es el tipo de señal en que se puede ver este fenómeno con mayor claridad. En tiempo, el objetivo es apreciar como el *dither* introduce "basculamientos" que deshacen los pulsos rectangulares que se forman al cuantificar, y en frecuencia, observar como las réplicas del tono, los armónicos, que son los causantes de la distorsión, desaparecen a cambio de aumentar el nivel de ruido en todo el espectro.

Tipos de dither y uso

Se presentan los tres tipos fundamentales de *dither* que existen en función del modelo estadístico que los genera, que son uniforme, triangular y Gauss. Se aportan los cálculos para deducir la potencia de ruido que introducen en la señal y

los parámetros estadísticos óptimos para lograr el mejor resultado, para de esta manera, discutir la influencia de cada uno de ellos en función de cómo “decorrelan” el ruido respecto a la señal.

Estudio práctico

Se presenta el applet con un estudio práctico con varios experimentos.

El primer experimento tiene como finalidad estudiar la acción del *dither* sobre un tono. Para ello, se debe cuantificar con un número de bits muy bajo (3) y comparar, subjetivamente, la audición del mismo. Sin *dither*, se escuchará el tono distorsionado, con *dither*, se escuchará el tono original pero acompañado de cierta cantidad de ruido blanco. En las gráficas, el usuario deberá observar como en tiempo, los pulsos rectangulares provocados por la cuantificación se “aleatorizan”, mientras que en el espectro, las réplicas del tono (armónicos) se transforman en una banda de ruido blanco esparcido en todo el espectro.

El segundo experimento tiene como objetivo estudiar el efecto del *dither* variando la potencia del mismo. Empleando una potencia inferior a la óptima, se observará como los armónicos desaparecen en parte, pero no totalmente. Con potencia óptima, desaparecen totalmente introduciendo la cantidad mínima de *dither*. Con potencia superior, los armónicos también desaparecen, pero se está añadiendo una cantidad de *dither* en exceso de forma innecesaria, lo que empeora la calidad de la escucha.

El tercer experimento está diseñado para apreciar, de forma subjetiva, el efecto auditivo de los diferentes tipos de *dither* que se pueden emplear. El *dither* uniforme linealiza la media, el triangular, la media y la varianza, y el Gauss, los momentos superiores al segundo. En la práctica, el oído humano es capaz de apreciar la linealización de media y varianza, el usuario deberá apreciar este hecho y de este modo, entender porqué el triangular es el tipo que se recomienda utilizar en equipos de audio. La linealización de la varianza deberá apreciarse con claridad al comparar fragmentos de silencio con aquellos en los que hay presencia de señal, observando cómo con *dither* uniforme, existe correlación entre señal y ruido de cuantificación.

El cuarto experimento tiene como objetivo estudiar su efecto para señales de voz y música con un número de bits de cuantificación alto (14 o 12), para de esta manera, apreciar su acción en casos que se aproximan en buena medida al paso de 24 bits a 16, que se realiza en audio profesional. La diferencia es en este caso muy sutil y entra en juego la capacidad auditiva del usuario.

Noise-Shaping

Introducción

Se explica el funcionamiento básico de la técnica de *Noise-Shaping* y se presenta un diagrama de un sistema de primer orden y la función de transferencia del mismo.

Análisis

Se explica, en detalle, el mecanismo de funcionamiento del *Noise-Shaping*, desarrollando la demostración matemática que da lugar a la función de transferencia que caracteriza un filtro de *Noise-Shaping* de orden N. Con ella, se indica el sentido de los elementos que intervienen y qué papel juegan, en concreto, los retardos y el orden de la función.

Desarrollo de $H(z)$

Se desarrolla la función obtenida antes para varios órdenes, presentando para cada uno, su expresión matemática y el diagrama de bloques del sistema que la implementa. Finalmente, se muestra un gráfico comparativo en el que se superponen las funciones de transferencia para varios órdenes, y de este modo, apreciar como a mayor orden, el desplazamiento de ruido a altas frecuencias es superior.

Modelado psicoacústico

En este apartado se exponen las técnicas de *Noise-Shaping* avanzadas que emplean el modelo psicoacústico del oído para mejorar la eficacia de este sistema, las cuales, son las líneas de investigación más recientes que se siguen en las técnicas de digitalización de audio. El objetivo es dar una visión general de las mismas, para dar a conocer que posibilidades se abren en este campo y cuáles son los logros hasta la actualidad. En primer lugar, se hace una breve presentación introductoria de los conceptos básicos del modelo psicoacústico del oído, desde un

punto de vista general, para poder entender la acción de estas técnicas. Luego, se explica, en concreto, como aprovechan estos sistemas esta propiedad del oído y que vías se han propuesto para resolver su implementación, pues no es una tarea trivial, ya que la resolución de las ecuaciones que aparecen es muy compleja. Por ello, se presentan cuales son las opciones que se manejan actualmente y cómo funcionan, a nivel elemental, discutiéndose las ventajas e inconvenientes de cada una y su viabilidad.

Sistema completo

Se presenta un diagrama de bloques del sistema que combina la aplicación de las tres técnicas estudiadas en el tutorial: sobremuestreo, *dither* y *Noise-Shaping*. Dicho sistema se puede considerar la meta final a la que se llega tras el estudio de estas técnicas, puesto que en la práctica, en los sistemas de audio reales, se suelen emplear de forma conjunta. Se aportan una serie de datos numéricos para comprobar de forma cuantitativa hasta qué punto se puede llegar a reducir el ruido de cuantificación con estas técnicas cuando son aplicadas de forma combinada y configurando cada parámetro de forma óptima.

Además, se hace un breve recordatorio del funcionamiento que tenían el sobremuestreo y el *dither*, para evitar que el usuario tenga que volver a capítulos anteriores si necesita revisar esos conceptos, y lo que es fundamental, como conclusión, se discute el papel que juega cada uno en el sistema conjunto y cuál es el efecto de combinarlos con el *Noise-Shaping*.

Estudio práctico

Se presenta el applet con un estudio práctico con varios experimentos.

El primer experimento está diseñado para estudiar el efecto de la técnica de *Noise-Shaping* dependiendo del orden de la función aplicada. Para una configuración estándar (14 bits de cuantificación, sobremuestreo por factor 4 y *dither* triangular óptimo), la cual da como resultado ruido de cuantificación muy aproximado al caso ideal de ruido blanco. Con ello, se observará la acción del filtro de *Noise-Shaping* para órdenes 1, 2 y 3. Se constatará cómo, a mayor orden, se desplaza más cantidad de ruido fuera del rango audible. En la gráfica de espectro

promedio, se observará, consecuentemente, cómo a mayor orden, la pendiente de la función es mayor.

El segundo experimento propone aplicar el sistema de *Noise-Shaping* eliminando la etapa de sobremuestreo, es decir, no aplicar interpolación previa, por lo que la frecuencia de muestreo de la señal no es modificada. Esta operación tiene como consecuencia el desplazamiento del ruido de cuantificación a la zona de alta frecuencia del espectro, pero todo él permanece dentro del rango audible, por lo que se percibe más agudo, pero no se reduce la potencia de éste. Este experimento tiene como objetivo resaltar la función de la etapa de sobremuestreo y a partir de ello, además, plantear la técnica de *Noise-Shaping* psicoacústico, en la cual, en cambio, la etapa de sobremuestreo no tiene sentido, ya que su objetivo es moldear el ruido de cuantificación dentro del rango audible de manera óptima, en vez de desplazarlo fuera.

4.2 Aplicaciones web

El tutorial de digitalización de audio contiene cuatro aplicaciones web que aportan una experiencia práctica para reforzar el aprendizaje del alumno.

Como se ha explicado, el objetivo es que el alumnos adquieran los conocimientos teóricos necesarios (por diferentes vías, como se ha indicado en capítulos previos) y que después, comprueben su aplicación y los resultados que ofrecen en la práctica.

Las aplicaciones web consisten fundamentalmente en la simulación de las técnicas estudiadas a través de implementaciones de las mismas sobre Java, de forma interactiva (con parámetros configurables), y del modo más sencillo, en medida de lo posible, para maximizar el rendimiento y el tiempo dedicado por el alumno a ellas.

La finalidad de las applets, hay que insistir, es puramente didáctica y han sido creadas para un uso ceñido estrictamente al ámbito académico (tanto universitario, como, eventualmente, personal). Tanto en lo referente al manejo de las applets, como los resultados que ofrecen y los algoritmos que implementan, no se han optimizado para que puedan ser utilizadas como herramienta de ayuda

profesional o semiprofesional en la creación de sistemas de audio. Es decir, que los resultados que aportan, tanto objetivos (valores numéricos del aumento o reducción de la SNR) como subjetivos (calidad del audio escuchado) son meramente orientativos y no deben tomarse como una regla totalmente fiable (aunque no obstante, se ha perseguido crear una algoritmia cuidada que aporte unos resultados lo suficientemente realistas para que puedan ser considerados una buena aproximación).

Las cuatro applets están dedicadas a cada uno de los cuatro grandes bloques que componen el tutorial de digitalización de audio:

- Cuantificación
- Oversampling (sobremuestreo)
- Dither
- Noise – Shaping (sistema completo)

Las applets contienen elementos que se solapan y de hecho, podrían haberse presentado fusionadas en una sola aplicación. Se ha optado, no obstante, por crear cuatro aplicaciones totalmente independientes para que las técnicas sean estudiadas con mayor claridad de forma individual y evitar confusiones o distracciones por parte del alumno, haciéndole seguir un camino de estudio determinado, marcado por el profesor.

Así, por ejemplo, el applet de sobremuestreo funciona simulando la “recuantificación” de una señal de audio aplicando dicha técnica y sin aplicar ninguna, para realizar una comparación entre ambas. Por tanto, el applet de cuantificación puede considerarse una parte extraída del applet de sobremuestreo, ya que ofrece la simulación del proceso de “recuantificación” sin aplicar ninguna técnica. La motivación de esta manera de presentar la parte práctica es que la filosofía del tutorial es guiar al alumno en el proceso de aprendizaje, de modo que se otorgue la relevancia que le corresponde a cada tema tratado. Es decir, que por ejemplo, en el caso concreto mencionado, que el número de bits de cuantificación no quede como un parámetro secundario, que pudiera ser ignorado por el alumno,

si no que merece ser estudiado con una aplicación creada específicamente para ello.

Asimismo, el applet de *Noise-Shaping* (la más compleja de todas, con diferencia) no está dedicado simplemente a estudiar esta técnica, si no que ofrece la simulación de un sistema en la que se aplica de forma conjunta con el sobremuestreo y el *dither*. Los sistemas de audio reales, en la práctica, son muy complejos e implementan algoritmos que combinan todas las técnicas estudiadas en el tutorial. Uno de los principales objetivos del mismo, es que el alumno estudie la acción e importancia de cada una de ellas sobre el resultado final si se aplican de manera combinada.

4.2.1 Descripción y funcionamiento de las applets

Se ha diseñado un esquema gráfico similar para todas las applets que forman parte del tutorial, agrupando las principales áreas funcionales en bloques similares, con parecida disposición y tamaño, para crear una interfaz coherente, intuitiva y usable.

4.2.1.1 Disposición gráfica de los bloques principales

1. Cabecera

En la cabecera de la aplicación, que consiste en una barra de horizontal situada en la parte superior, se sitúan los controles para cargar un archivo de audio, reproducirlo y mostrar la información referente al mismo.

2. Cuerpo

El cuerpo es la parte principal de la aplicación, ocupa la mayor parte del espacio. En él se sitúan los controles para modificar los parámetros configurables de la técnica estudiada y se muestran los resultados.

2.1. Área de controles de configuración de parámetros

2.2. Área de resultados

2.2.1. Zona de resultados gráficos

2.2.2. Zona de resultados numéricos

2.2.3. Zona de resultados auditivos (controles de reproducción de audio)

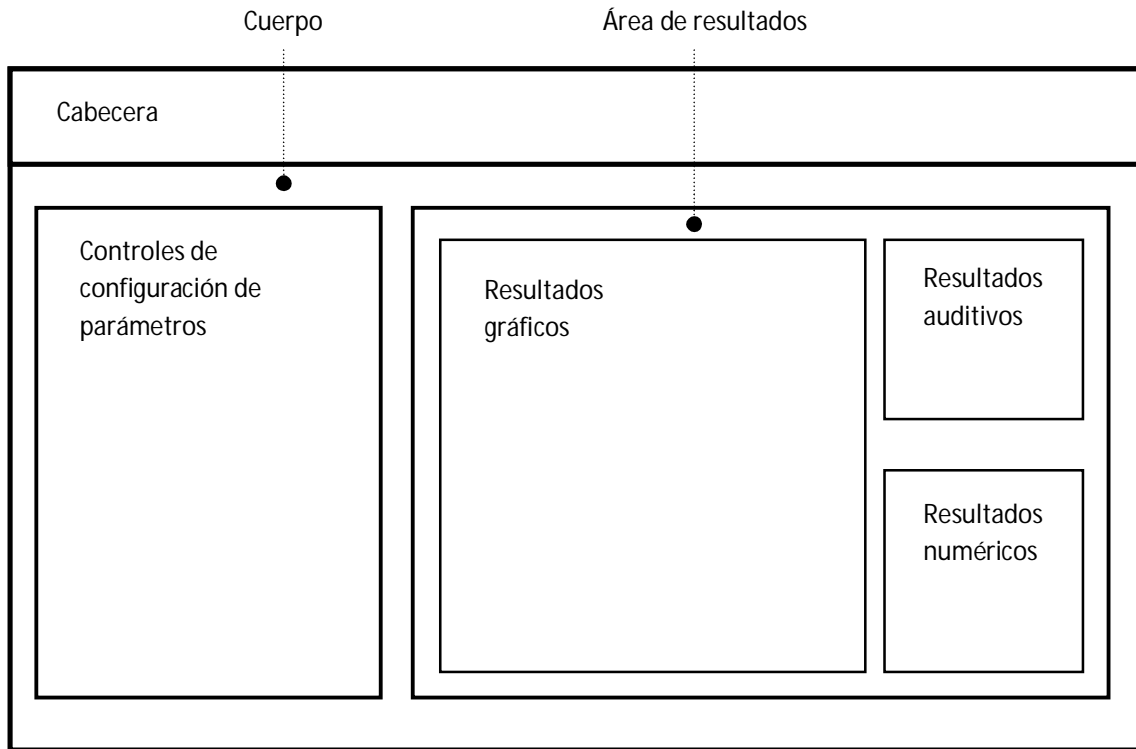


Figura 4-9 Estructura básica de las aplicaciones

El diseño realizado procura presentar al usuario la aplicación de manera que se pueda utilizar de un modo intuitivo y natural, de forma que encuentre los diferentes elementos en un lugar lógico, acorde a la secuencia de pasos que se deben dar para realizar una práctica. Así, lo primero que se encuentra, al inspeccionar el applet de arriba a abajo, es la cabecera, para realizar la acción inicial, elegir un fichero de audio. Luego, una vez se ha pasado al cuerpo, de izquierda a derecha, primero se presentan los elementos de configuración y luego los de resultados. Es decir, se sigue el orden natural de lectura de un texto. Además, para reforzar este propósito, los elementos aparecerán de forma progresiva. Cuando se inicia la aplicación, algunos de los elementos están ocultos, para evitar distraer la atención del usuario, y se muestran cuando éste llega al punto en que es necesaria su presencia.

Secuencia de acciones genérica, común a las cuatro applets:

1. Carga de un archivo de audio y reproducción del mismo
2. Configuración de parámetros y aplicación de la técnica
3. Observación de los resultados: gráficos, numéricos y auditivos
4. Vuelta al punto 1 o al punto 2

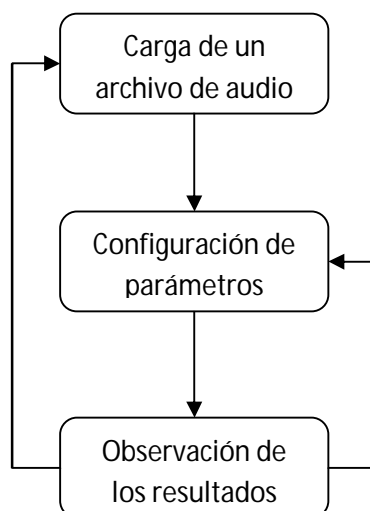


Figura 4-10 Secuencia básica de manejo

4.2.1.2 Elementos y controles de las applets

Los elementos y controles que forman las applets se pueden dividir en dos tipos: los comunes y los propios de cada applet. Los elementos comunes son la cabecera y el panel de resultados, los elementos específicos corresponden al área de controles de configuración, puesto que cada técnica tiene sus propios parámetros.

4.2.1.2.1 Cabecera

La cabecera tiene como finalidad ofrecer una barra de herramientas con las opciones de abrir un fichero de audio y reproducirlo, así como mostrar información relativa al mismo.

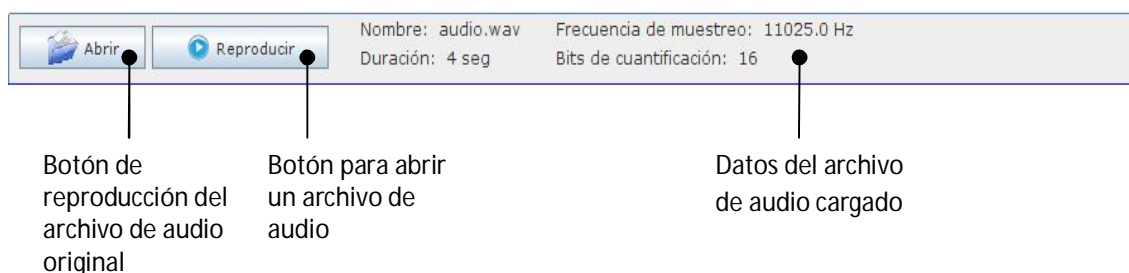


Figura 4-11 Cabecera

El botón de Abrir, al ser pulsado, da lugar a la apertura de un cuadro de diálogo en el que se ofrecen diferentes opciones para obtener el fichero:

- Carga de un ejemplo preparado especialmente para su uso en el applet
- Carga de un fichero local
- Carga de un fichero en el repositorio introduciendo su nombre.

Se ofrecen diversas opciones para cargar el fichero de audio para aumentar la versatilidad del tutorial. Si bien lo recomendado por el profesor será siempre utilizar los ejemplos preparados especialmente para observar la aplicación de las técnicas, se deja la opción de cargar otros ficheros a través de diferentes vías, para que, por ejemplo, el alumno pueda probar las técnicas con ficheros de audio de su interés o en el futuro, el profesor pueda añadir más material de trabajo sin necesidad de modificar el código fuente.



Figura 4-12 Cuadro de diálogo "Abrir fichero de audio"

El objetivo es que las applets sean utilizadas con fragmentos de audio cortos, de cinco a quince segundos, duración suficiente para apreciar la acción de las técnicas.

El funcionamiento de la carga del fichero de audio es guiado. Cuando el applet se carga, el botón reproducir aparece desactivado. El usuario debe elegir, obligatoriamente, un fichero de audio. Una vez se ha realizado esta operación, aparecen los datos del mismo y el botón reproducir pasa a estado activo.

El botón Abrir se puede pulsar cuantas veces se quiera. Cada vez que se cargue un fichero, supone el reinicio de la aplicación, eliminándose todos los resultados que puedan haber aparecido durante el uso del applet con un fichero de audio anterior.



Figura 4-13 Secuencia de apertura de un fichero de audio

4.2.1.2.2 Área de resultados

El área de resultados está dividido en tres partes, mostrando las herramientas de análisis que se pueden emplear para observar la acción de las técnicas.

La presentación de los resultados de salida es una parte de las applets que, teniendo en cuenta que se trata de un tutorial con fines didácticos, puede plantearse de varias maneras.

La disyuntiva surge entre mostrar los resultados de forma guiada, es decir, a través de un proceso secuencial en el que el alumno vaya pulsando botones de avance, o bien, mostrar todos los resultados de forma inmediata en un panel combinado, de modo que el alumno decida por sí mismo.

Se ha optado por la segunda opción, para que las applets se puedan utilizar con la mayor libertad posible, ya que es muy útil tener todos los datos presentados de una manera agradable y cómoda en una misma vista.

Se presentan tres tipos de resultados que se indican a continuación.

4.2.1.2.2.1 Gráficas de tiempo y frecuencia

Se muestra una serie de gráficas de la señal en tiempo y de su densidad espectral de potencia (frecuencia), correspondientes a puntos concretos de interés del sistema implementado. Las gráficas se agrupan en un panel de pestañas que el usuario puede seleccionar y comparar.

Las pestañas aparecen de forma secuencial, de modo que cuando el usuario realiza acciones que dan lugar a salidas del sistema, nuevas pestañas se van añadiendo. Así, por ejemplo, cuando se carga la señal de audio, aparece la pestaña correspondiente a la señal de entrada (x). Cuando realiza la cuantificación,

aparecen las pestañas correspondientes a ésta (yA, eA). Cuando aplica la técnica estudiada, aparecen la serie de pestañas que muestran los gráficos de la señal en puntos intermedios de interés del sistema estudiado y finalmente, la salida.

El panel de gráficas se divide en dos partes. En la parte superior, aparece la gráfica en tiempo de la señal, en un fragmento de medio segundo. Mediante dos botones de avance y retroceso, se puede recorrer toda la señal para estudiar el instante que interese. En la parte inferior, aparece un gráfico de la densidad espectral de potencia de una ventana de la señal, de aproximadamente, 20 ms. Mediante un slider situado en la parte inferior de la gráfica de tiempo, se puede seleccionar el instante en el cual se desea centrar la ventana. La actualización de la DEP es instantánea, mientras se desplaza el slider a lo largo del tiempo, se puede observar como varía la DEP.

Las gráficas ofrecen la funcionalidad de zoom, que se utiliza dibujando un área rectangular sobre la misma, sobre la cual, se realizará una ampliación.

Se ofrece también un menú contextual, pulsando botón derecho, que muestra varias acciones, como guardar las gráficas en un archivo PNG, resetear el zoom para volver a la escala normal, y modificar el estilo de representación.

Las gráficas pueden ser de dos tipos: simples y múltiples, tanto en tiempo como en frecuencia. Las gráficas simples muestran una sola señal, las gráficas múltiples pintan dos señales superpuestas, correspondientes a puntos equivalentes de los dos sistemas estudiados, para realizar una comparación rápida y directa entre ambos. La correspondencia se indica mediante colores, el esquema de cada sistema a estudiar está enmarcado en un color que se corresponde con el de la señal obtenida (en general, rojo para la señal cuantificada y azul, para la señal cuantificada aplicando alguna técnica).

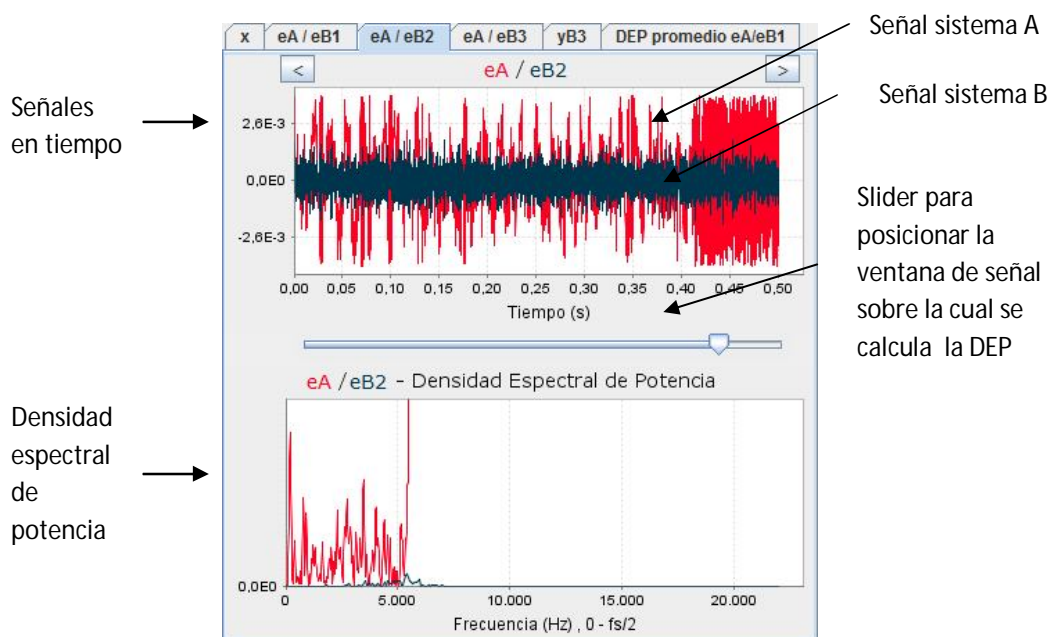


Figura 4-14 Resultados gráficos

En este ejemplo, el alumno puede comparar la señal sometida a recuantificación (color rojo) frente a la señal a la que se le aplica *Noise-Shaping* (color azul), comprobando que, efectivamente, se reduce el nivel de ruido de cuantificación.

4.2.1.2.2.2 Resultados de audio

Se ofrecen las señales de audio obtenidas tras aplicar la técnica estudiada frente a la señal recuantificada sin aplicar ninguna técnica. El objetivo de esta opción es evaluar subjetivamente la acción de las técnicas, comprobando que, efectivamente, la aplicación de la técnica estudiada supone una mejora en la calidad de escucha del audio.

Hay que resaltar que en este método de observación es importante apreciar el efecto negativo de escuchar la distorsión provocada por la cuantificación y la acción de la técnica del *dither*, cuyo efecto es decorrelar señal y ruido, transformando éste en ruido blanco uniforme. Esta es una sensación subjetiva que no puede ser evaluada cuantitativamente, mediante resultados numéricos o gráficas, por lo que la escucha del audio es esencial para comprobar este fenómeno.

Por cada punto de interés se ofrece señal y error. La escucha del error separado de la señal (obtenido de la resta de la señal cuantificada menos la señal original) ayuda a apreciar con mayor nitidez la reducción del nivel de ruido de cuantificación y la transformación de la distorsión en ruido blanco uniforme gracias al *dither*.

La escucha de las señales de audio se realiza pulsando el botón correspondiente, el cual reproduce la señal entera hasta que termina. No se ofrecen controles de reproducción, como pausa, rebobinado, etc, para simplificar la interfaz, considerando que se van a utilizar fragmentos de audio cortos, de cinco a quince segundos, por lo que la idea es escucharlos enteros y repetirlos cuantas veces sea necesario.

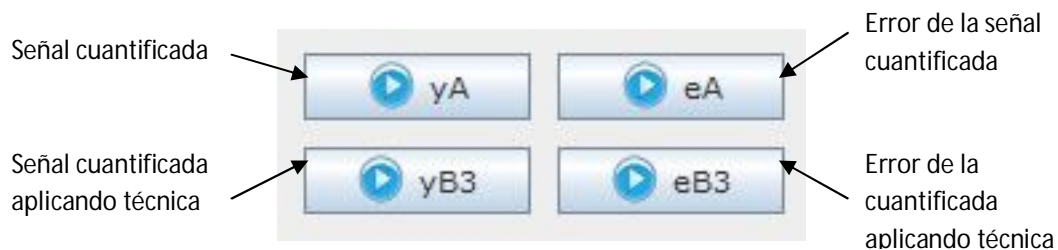
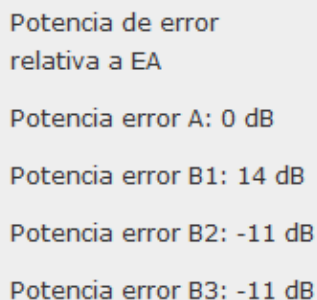


Figura 4-15 Resultados auditivos

4.2.1.2.2.3 Resultados numéricos

Se ofrece un panel de resultados en el que se indican los valores de potencia del error de cuantificación en las señales de salida de los sistemas estudiados, así como su valor en diversos puntos intermedios de interés de los mismos.

Los resultados se expresan en decibelios de forma relativa al error de de la señal recuantificada sin aplicar ningún sistema (el sistema A). Si el resultado es negativo, significará que se ha producido una reducción del error de cuantificación y si es positivo, significará que ha aumentado.



Potencia de error
relativa a EA

Potencia error A: 0 dB

Potencia error B1: 14 dB

Potencia error B2: -11 dB

Potencia error B3: -11 dB

Figura 4-16 Resultados numéricos

La potencia del error de cuantificación se calcula como:

$$E_{dB} = 10 \log_{10} \left(\frac{1}{N} \sum_{n=0}^{N-1} |e[n]|^2 \right)$$

Siendo el error $e[n]$ la diferencia en cada muestra entre la señal original y la señal en el punto considerado del sistema estudiado

$$e[n] = x[n] - x'[n]$$

4.2.1.2.3 Área de configuración de parámetros y aplicación de la técnica

El área de configuración contiene una serie de controles, como cajas desplegadas y campos de texto, en el que el alumno puede modificar los valores de los parámetros que intervienen en la aplicación de las técnicas estudiadas, para observar el efecto de cada uno de ellos sobre el resultado del sistema.

Junto a ellos, aparecen los botones que aplican la técnica estudiada una vez el alumno ha configurado dichos parámetros y que lanzan el procesamiento de la señal de audio, para obtener los resultados gráficos, de audio y numéricos explicados antes.

En todas las applets, con excepción de la de cuantificación, el proceso se realiza siempre en dos pasos:

1. Configuración del número bits de cuantificación y aplicación
2. Configuración de los parámetros de la técnica estudiada y aplicación

En primer lugar, el alumno establece un número de bits con el que la señal será recuantificada, y pulsa el botón “Aplicar cuantificación”, escuchando inmediatamente el resultado, de forma que compruebe la reducción de la SNR y la aparición de distorsión debida a la recuantificación.

En segundo lugar, el alumno configura la serie de parámetros que correspondan a la técnica estudiada (sobremuestreo, *dither* o *Noise-Shaping*) y pulsa el botón “Aplicar <<nombre de la técnica>>”, escuchando inmediatamente la señal resultado de aplicar la técnica cuantificando con el número de bits que había establecido en el punto 1, para observar el efecto, con las herramientas de análisis explicadas antes, que en general, será una reducción del ruido de cuantificación (aunque no necesariamente).

Realizado el proceso, el alumno puede cambiar los parámetros y aplicar de nuevo la técnica, o bien, empezar de nuevo con un número de bits de cuantificación diferente, o incluso, cambiar el fichero de audio.

El applet de *Noise-Shaping* contiene un panel de parámetros que recoge todas las opciones posibles, pues es la aplicación en la que finalmente se implementa un sistema que combina las tres técnicas estudiadas, para observar la influencia de cada una. El resto de applets, por tanto, tienen un panel de parámetros con sólo algunos de los que incluye el applet de *Noise-Shaping*.

Paso 1

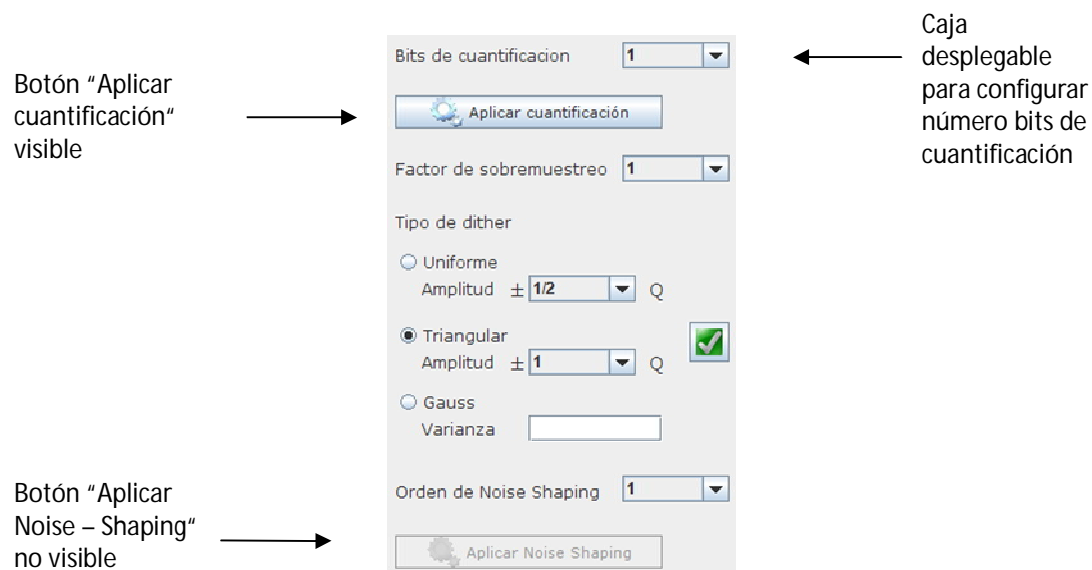


Figura 4-17 Configuración de parámetros (Paso 1)

Paso 2

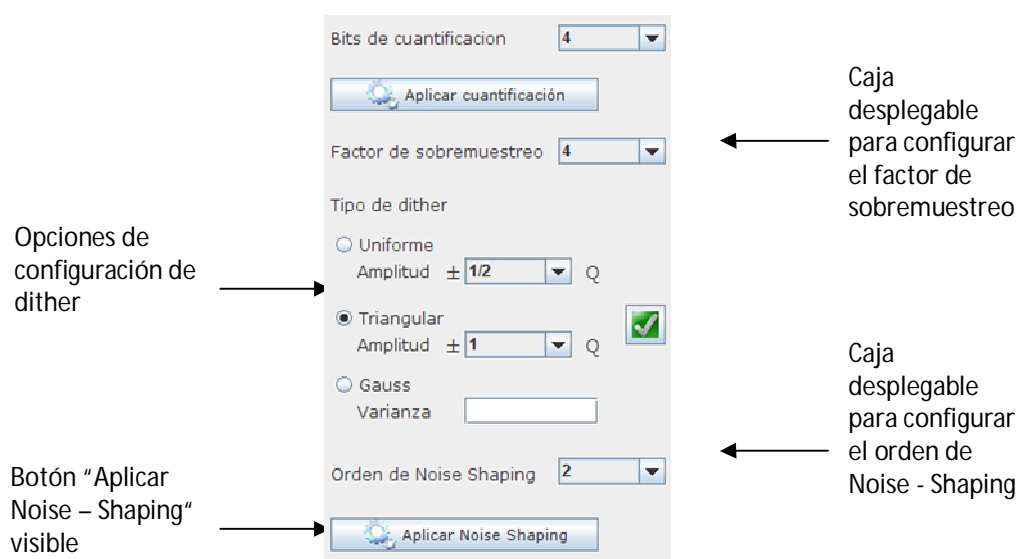


Figura 4-18 Configuración de parámetros (Paso 2)

4.2.2 Vista completa de las applets

Applet de cuantificación

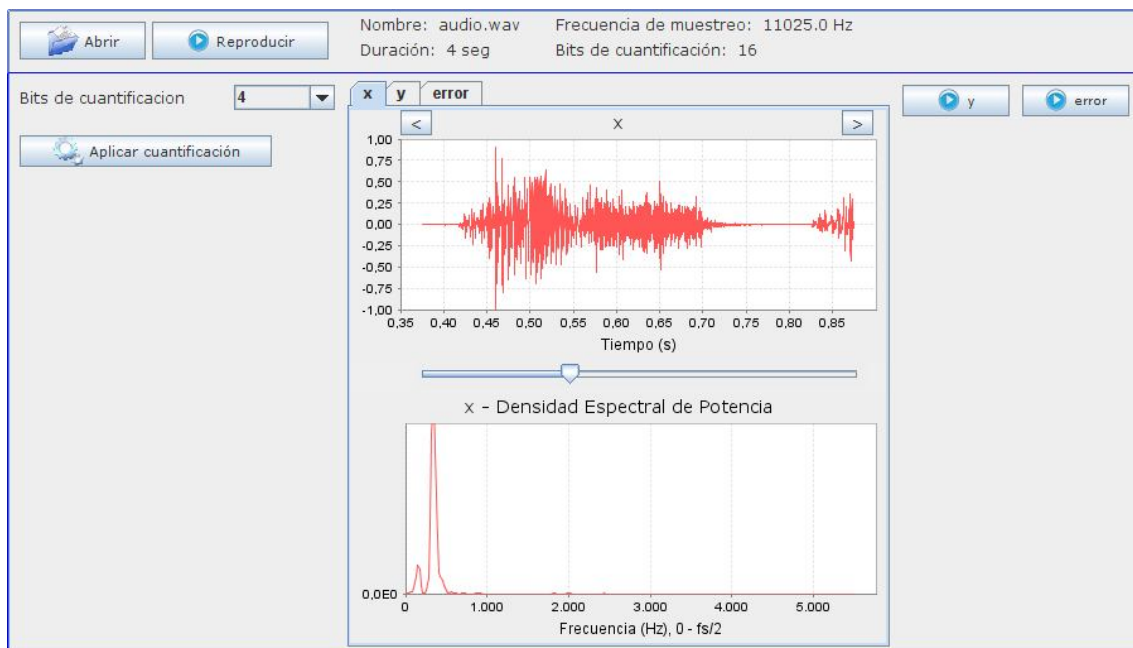


Figura 4-19 Vista principal del applet de cuantificación

En el applet de cuantificación no aparecen resultados numéricos pues estos se refieren siempre a la diferencia entre aplicar una técnica y cuantificar de forma simple, por lo que en este caso, no hay ningún valor que mostrar.

Applet de sobremuestreo

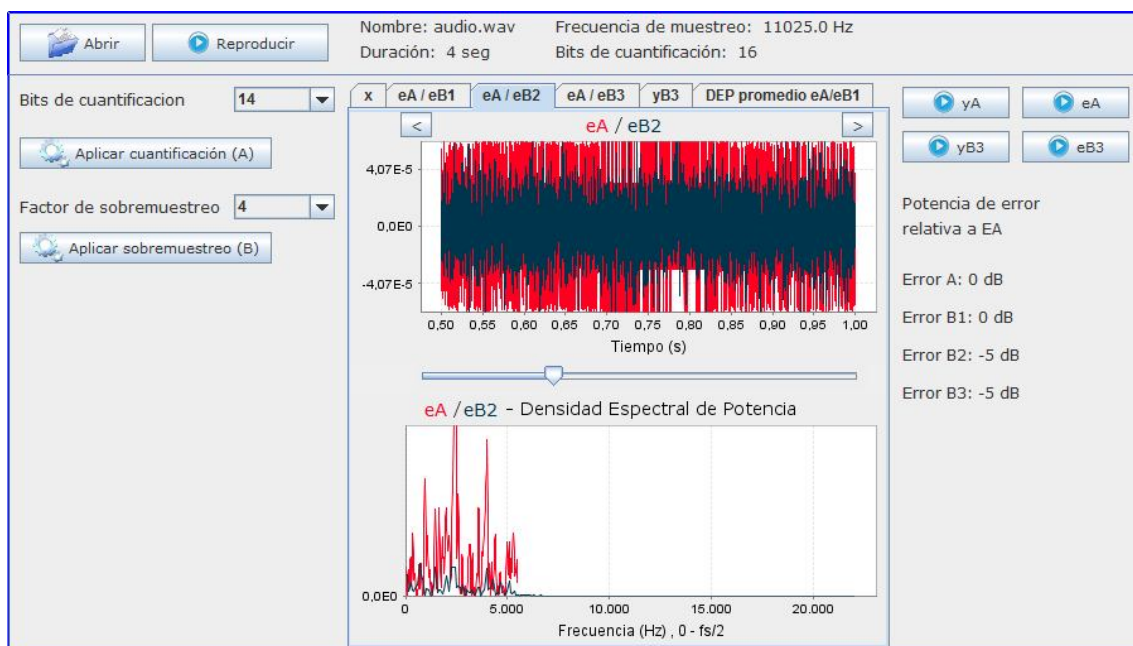


Figura 4-20 Vista principal del applet de sobremuestreo

Applet de dither



Figura 4-21 Vista principal del applet de dither

Applet de Noise-Shaping

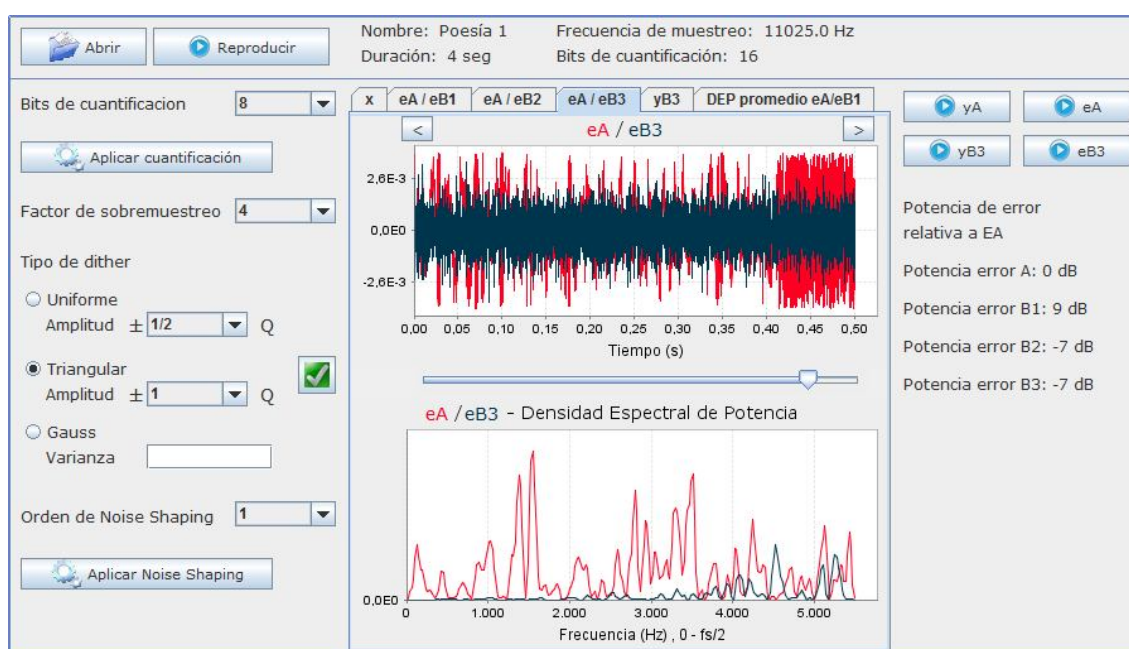


Figura 4-22 Vista principal del applet de Noise - Shaping

4.2.3 Uso aplicado de las applets

Las applets ofrecen una interfaz en la que el alumno puede configurar los parámetros de las técnicas, observar los resultados con total libertad y experimentar y hacer pruebas como considere conveniente. No obstante, existen una serie de situaciones para cada técnica en las que se puede apreciar su acción con especial claridad. Es muy recomendable, por tanto, que en primer lugar, se observen estos casos para tomar conciencia en profundidad del funcionamiento de las técnicas, y se comprendan éstas con claridad, que es el objetivo fundamental del tutorial, para luego, si se desea, hacer pruebas libremente. En el tutorial estas situaciones vendrán indicadas como experimentos en la guía de uso que acompaña cada applet.

4.2.3.1 Applet de cuantificación

El applet de cuantificación se ha diseñado para apreciar de una forma muy simple el aumento del nivel de ruido de cuantificación conforme se reduce el número de bits empleado, en concreto, seis decibelios más por cada bit menos utilizado.

El intervalo de valores de bits de cuantificación que se ofrece está entre 1 y 16. El máximo corresponde al número de bits del estándar CD, del cual, procederán las señales de audio que se emplearán en el tutorial.

Una posible situación de estudio es observar el resultado de cuantificar con 12, 8 y 4 bits una señal de voz. En el primer caso, se apreciará la aparición de un ligero ruido de cuantificación, muy aproximado a la hipótesis de ruido con fdp uniforme y blanco, que no degrada excesivamente la SNR pero que, evidentemente, provoca cierta molestia. En el segundo caso (un valor, cabe señalar, estándar en algunos formatos de audio de baja calidad) ya se observa la aparición de distorsión y el ruido alcanza un nivel notorio que dificulta notablemente la escucha. El tercer caso, se obtendrá una señal de audio muy degradada, con un nivel de ruido de cuantificación excesivo, en la cual, la distorsión es muy fuerte.

4.2.3.2 Applet de sobremuestreo

El applet de sobremuestreo expone, básicamente, la reducción del nivel de ruido de cuantificación de forma proporcional al factor de sobremuestreo.

El procedimiento ideal para observar la acción de la técnica es recuantificar una señal de voz con un número de bits no excesivamente bajo, como 14 o 12 (pulsando el botón "Aplicar cuantificación)". El resultado será la aparición de ruido de cuantificación que se aproximará bastante a la hipótesis de ruido blanco y uniformemente distribuido. Luego, se procederá a realizar la recuantificación aplicando sobremuestreo, en primer lugar, por un factor 2, en segundo lugar, por un factor 4 y finalmente, por un factor 8. Se observará que duplicando el factor de sobremuestreo, el nivel de ruido se reduce en 3 decibelios.

Una vez se ha comprobado la utilidad de la técnica de sobremuestreo, se analiza su funcionamiento observando las modificaciones que sufre la señal en cada etapa del sistema hasta obtener la salida.

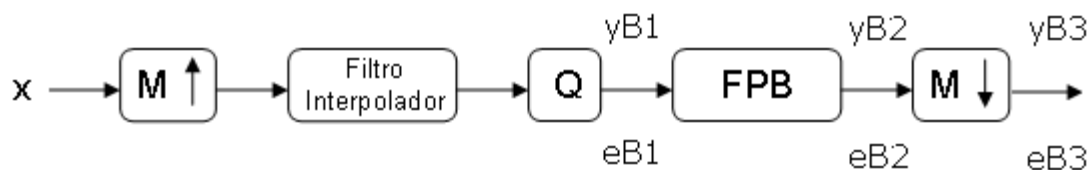


Figura 4-23 Diagrama de sobremuestreo

En el punto 1 (pestaña eA/eB1) se observa como el error de la señal sobremuestreada (color azul) se extiende en el espectro, su ancho de banda es igual a $f \cdot fs/2$ y su nivel se reduce de forma proporcional, pero la potencia de ruido sigue siendo la misma, ya que el área de eA y eB1 se mantiene constante y en los resultados numéricos, por tanto, la diferencia de potencia de eB1 y eA es de 0 dB.

Es importante observar la diferencia entre los resultados para diferentes ventanas de la señal, que se obtienen desplazando el slider, y el promedio de todas las ventanas. Observando diferentes ventanas se aprecia que el ruido tiende a ser uniforme, analizando el promedio de todas ellas se comprueba que efectivamente, es así.

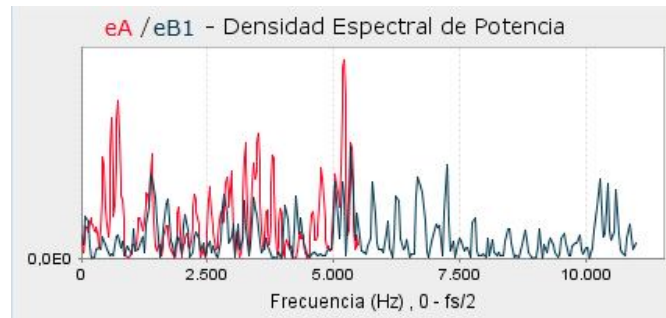


Figura 4-24 Gráfica de sobremuestreo (eA / eB1)

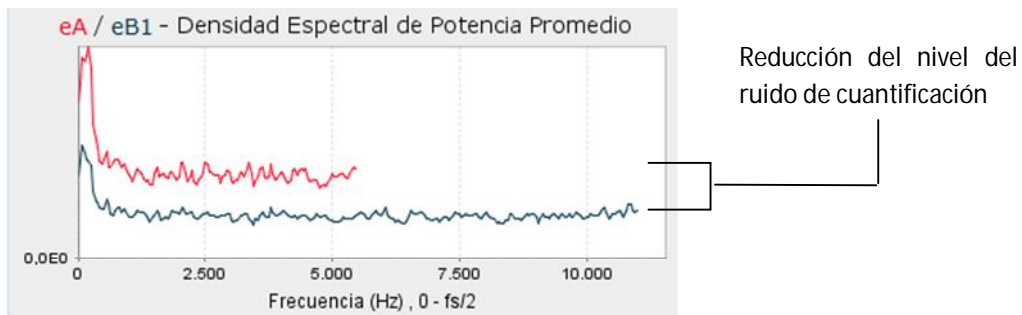


Figura 4-25 Gráfica de DEP Promedio de Sobremuestreo

En el punto 2 (pestaña eA/eB2), se observa la salida del filtro paso bajo que se aplica para eliminar la parte del espectro sobrante, que queda por encima de la frecuencia $fs/2$ (el ancho de banda original), y así poder, luego, realizar el diezmado sin que se produzca aliasing. Es en este punto donde se produce la reducción efectiva de la potencia de ruido de cuantificación. En la gráfica de la DEP, se apreciará como ha desaparecido la parte del espectro superior a $fs/2$ y en los resultados numéricos, la potencia de error eB2 relativa a eA será negativa, por ejemplo, - 3dB en el caso de haber empleado un factor 2.

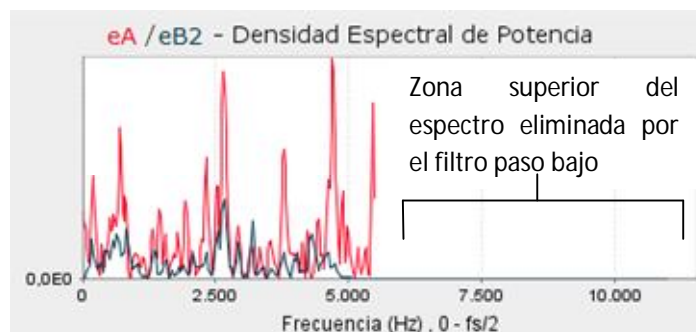


Figura 4-26 Gráfica de sobremuestreo (eA / eB2)

En el punto 3 (pestaña eA/eB3), finalmente, se observa como mediante el diezmado la señal eB2 vuelve a tener la frecuencia de muestreo original y se mantiene la reducción de ruido de cuantificación lograda.

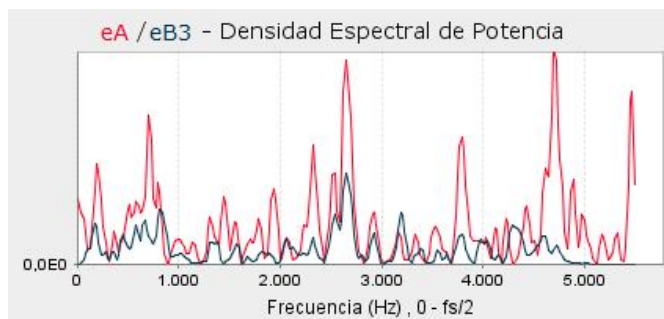


Figura 4-27 Gráfica de sobremuestreo (eA / eB3)

Una vez se ha estudiado el funcionamiento del sistema, se puede experimentar con la utilización de un número de bits de cuantificación inferior, 8 o incluso 4, para que la distorsión sea mucho mayor y no se cumpla la hipótesis de ruido uniforme y blanco. En ese caso, se observará que la efectividad de la técnica de sobremuestreo es menor y se elimina menos ruido de cuantificación.

Nº de bits empleado	Reducción de la potencia del ruido de cuantificación con factor 2
14	-3 dB
12	-4 dB
8	-2 dB
4	-2 dB

4.2.3.3 Applet de dither

El applet de *dither* tiene como objetivo apreciar cómo se puede eliminar la distorsión producida por la recuantificación de una señal sumando una señal aleatoria (ruido blanco) con una distribución adecuada.

Dicha distorsión consiste en la escucha de un desagradable efecto de granulado en el ruido de cuantificación. Sumando el *dither*, desaparece, pero a costa de pagar una penalización, un aumento de la potencia de ruido total. La razón de este fenómeno es que el oído percibe la distorsión cuando la media y la varianza del ruido de cuantificación no es nula, la finalidad del *dither* es linealizar dichos

parámetros (momentos de primer y segundo orden) para que el usuario no lo perciba.

El applet ofrece tres tipos diferentes de *dither* que emplear (en función de la densidad de probabilidad que caracteriza la señal aleatoria). El óptimo es el triangular, puesto que linealiza media y varianza. El uniforme solo linealiza la media, por lo que se observará como elimina la distorsión solo en parte. El ruido Gauss, por contra, elimina toda la distorsión (los momentos superiores), pero suma más ruido del mínimo necesario, por lo que, finalmente, se llega a la conclusión de que el *dither* óptimo es el triangular.

Para el *dither* uniforme y el triangular, se ofrece una lista de valores de amplitud respecto al escalón de cuantificación ($\pm 1/8 Q$, $\pm 1/4 Q$, $\pm 1/2 Q$, $\pm 1 Q$, $\pm 3/2 Q$) que pueden tomar. Para el uniforme el valor óptimo es $\pm 1/2 Q$ y para el triangular $\pm 1 Q$. En el caso del *dither* de tipo Gauss, se puede introducir directamente el valor de varianza que se desea emplear. Si se emplea valor de amplitud o varianza inferior al óptimo, se apreciará que se introduce menos ruido, pero no se produce una decorrelación completa del ruido de cuantificación respecto a la señal, por el contrario, si se emplea un valor superior, se estará sumando más ruido del mínimo necesario para conseguir transformar completamente el ruido de cuantificación en ruido blanco.

Se ofrece un botón (verde con una marca) que sirve para configurar los valores de *dither* óptimos para ayudar al usuario en caso de que no esté seguro de cuáles son.

Utilizando una señal de voz se puede apreciar con bastante claridad este efecto, y comprobar que, efectivamente, aunque al principio siempre resulte un concepto paradójico, la suma de ruido elimina la distorsión y mejora la escucha.

Con una señal de voz es sencillo apreciar el fenómeno, sin embargo, se puede estudiar en mayor profundidad si se utiliza como ejemplo una señal sinusoidal.

En el espectro, se observarán los armónicos que aparecen para la frecuencia concreta del tono empleado, viéndose como esto supone la concentración del ruido

en frecuencias muy concretas, hecho el cual provoca la distorsión. Aplicando el *dither*, el ruido de cuantificación se extiende de manera uniforme por todo el espectro, pero su nivel medio (la potencia total) aumenta.

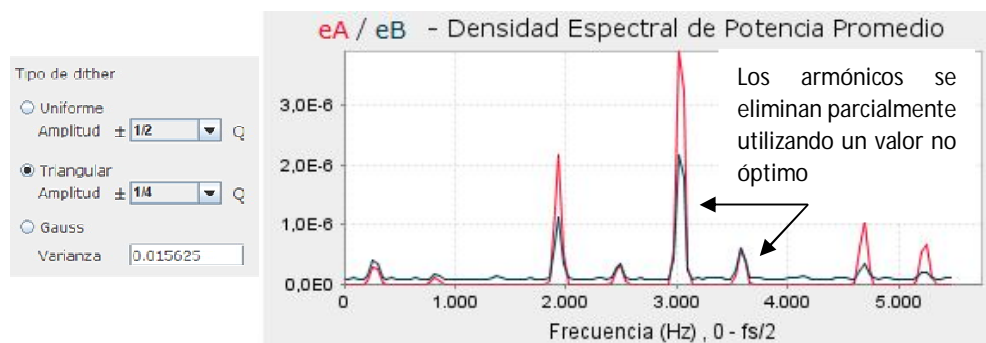


Figura 4-28 Aplicación no óptima de dither (en el espectro)

Empleando un valor de dither inferior al óptimo se observa como el ruido de cuantificación no se transforma totalmente en ruido blanco

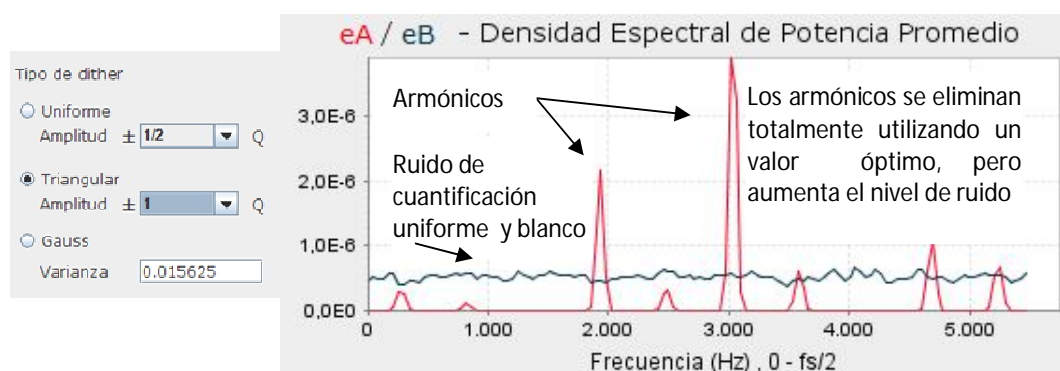


Figura 4-29 Aplicación óptima de dither (en el espectro)

Empleando un dither triangular y de valor ± 1 se observa como los armónicos que provocan distorsión desaparecen completamente, pero la potencia del ruido de cuantificación aumenta.

En el tiempo, se observa como el resultado de cuantificar una onda sinusoidal con un número de bits excesivamente bajo es que dicha onda cuantificada tome forma de pulsos cuadrados, que provocan la aparición de los armónicos observados antes, ya que su transformada de Fourier es una sinc. Aplicando el *dither*, se producen oscilaciones dentro de los pulsos, por lo que se aleatorizan y no aparecen, al transformarlos al espectro, los armónicos perjudiciales.

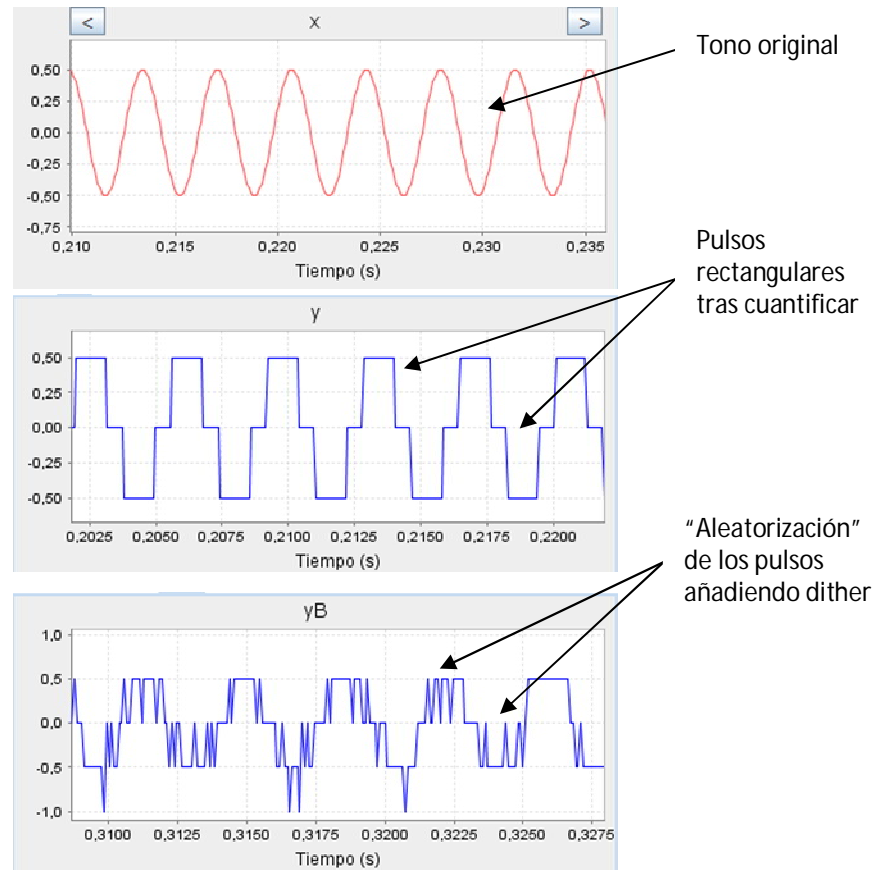


Figura 4-30 Efecto del dither en el tiempo

4.2.3.4 Applet de Noise – Shaping (sistema completo)

El applet de *Noise–Shaping* simula un sistema que combina la acción conjunta de las tres técnicas estudiadas: sobremuestreo, *dither* y *Noise–Shaping*.

El *dither* elimina la distorsión producida por la cuantificación a costa de aumentar el nivel de ruido. El sobremuestreo multiplica el ancho de banda de la señal y por tanto, extiende el ruido en el espectro, como se ha comentado antes, ya que el área se mantiene constante y por tanto, el nivel de ruido tiene que descender. Finalmente, la acción del *Noise–Shaping* consiste, esencialmente, en desplazar el ruido que queda dentro de la banda de audio a frecuencias superiores, para luego ser eliminado al realizar el diezmado.

Se pueden tomar una serie de valores de cuantificación para observar el funcionamiento de la técnica, por ejemplo, 12, 8 y 4. Para cada uno de ellos, se puede probar el efecto, sobre el resultado final, de modificar un parámetro (factor de sobremuestreo, modelo de *dither*, orden de *Noise–Shaping*). En general, será proporcional: cuanto mayores sean los valores del factor de sobremuestreo y de

orden de *Noise-Shaping* empleados, mas se reducirá el ruido de cuantificación. No obstante, en el caso de *Noise-Shaping*, existe una excepción, en ocasiones, dependiendo del rango dinámico de la señal de entrada y de los valores de amplitud entre los que oscile, puede convertirse en un sistema inestable, por lo que se obtendrá una señal incoherente. Esto sucede para valores de orden de *Noise-Shaping* elevados, pues cuanto mayor es el orden utilizado, mas ganancia se introduce, lo cual puede provocar que se rebasen los límites del fondo de escala, lo que hace que la realimentación se produzca de manera incorrecta y el sistema sea inestable.

El sistema de *Noise-Shaping* tiene una serie de etapas que deben ser observadas en el applet para comprender su funcionamiento, siguiendo la evolución del error de cuantificación.

En primer lugar, se observa la señal en el punto 2, pestaña eA/eB1. En ella se ve como la señal tiene un ancho de banda extendido, puesto que se ha sobremuestreado (proceso que se ha tenido que estudiar antes en el applet de sobremuestreo) y buena parte del ruido ha sido desplazado a las frecuencias superiores por la acción del filtro de *Noise-Shaping*.

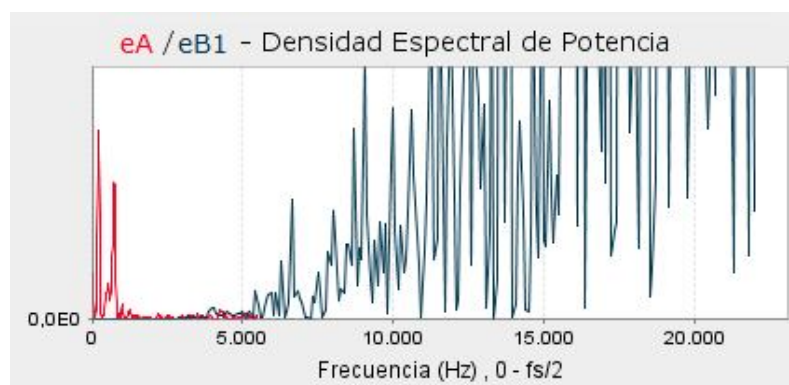


Figura 4-31 Gráfica de Noise - Shaping (eA / eB1) espectro

Desplazando el slider se apreciará la variabilidad en el tiempo de este efecto respecto a la señal de entrada. La gráfica del promedio en toda la señal del error eB1 mostrará como éste tiende a tomar la forma del filtro de *Noise-Shaping* que le moldea.

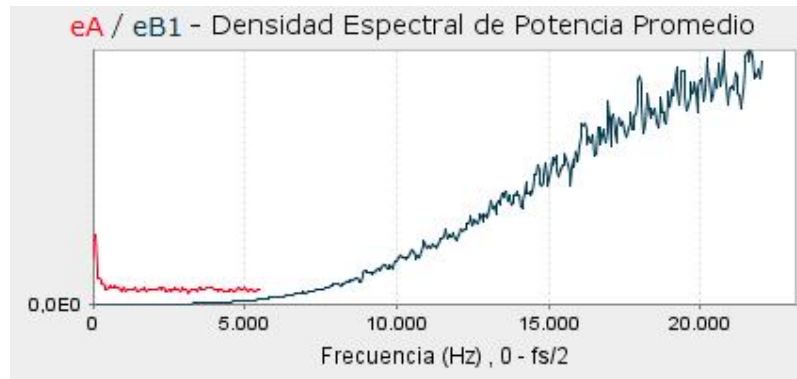


Figura 4-32 Gráfica de Noise - Shaping DEP Promedio (eA / eB1) en espectro

En el punto 2, pestaña eB2 / eA, se observa el resultado de aplicar el filtro paso bajo, el cual elimina todo el error de cuantificación que quede por encima de la frecuencia de muestreo.

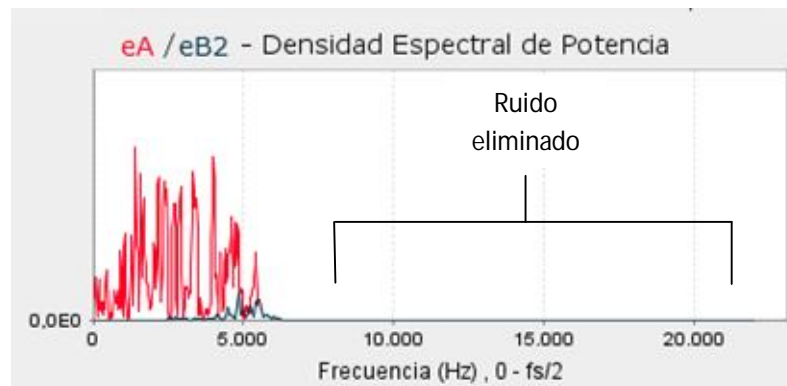


Figura 4-33 Gráfica de Noise - Shaping (eA / eB2) en espectro

En esta etapa ya se puede observar como en el tiempo, el ruido de cuantificación tiene una amplitud considerablemente menor.

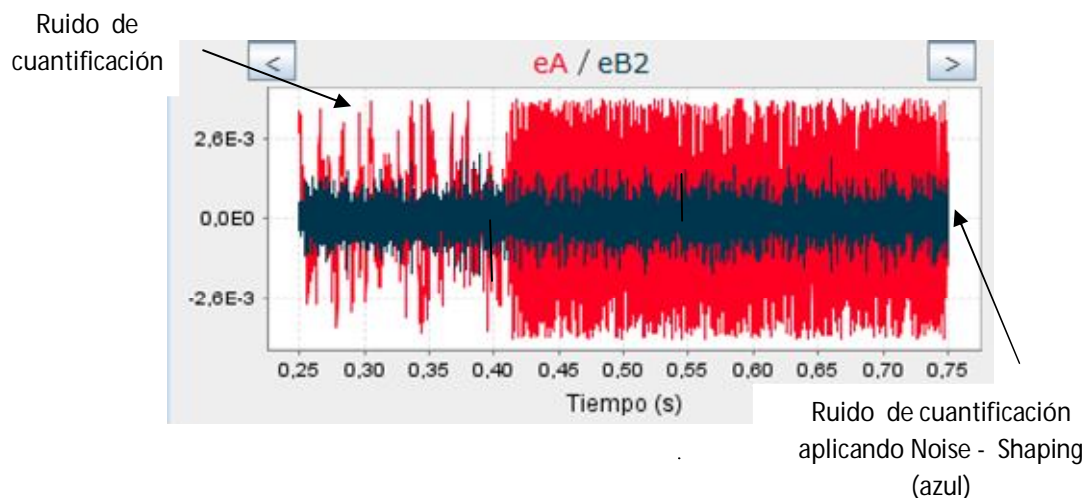


Figura 4-34 Gráfica de Noise - Shaping (eA / eB3) en tiempo

Finalmente, en el punto 4, pestaña eA / eB4, se realiza un diezmado para que la señal vuelva a tener la frecuencia de muestreo original, lo cual no altera el espectro de las señales, manteniéndose la rebaja de ruido de cuantificación conseguido antes.

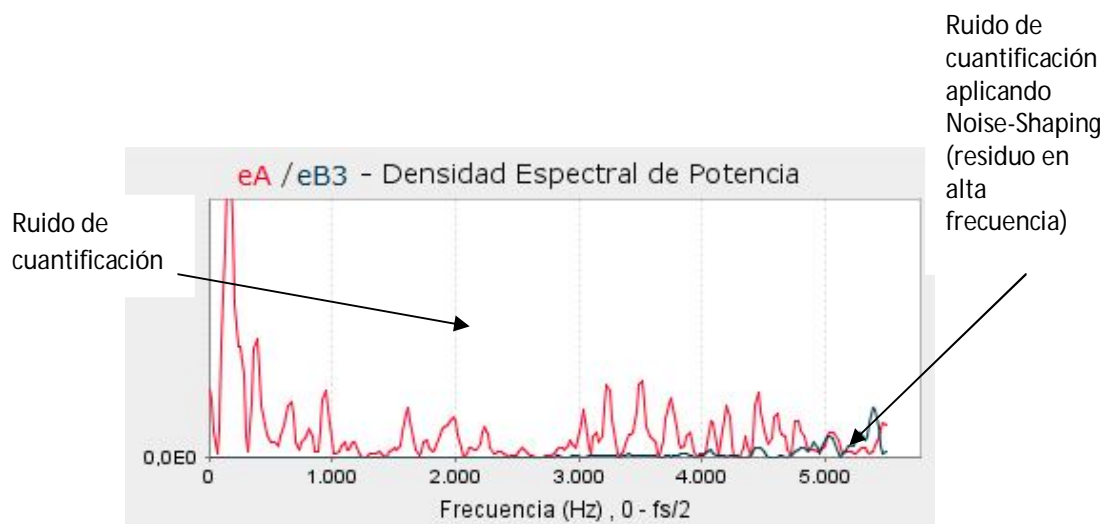


Figura 4-35 Gráfica de Noise-Shaping (eA / eB3) en espectro

Es importante ver que el filtro de *Noise-Shaping* desplaza a frecuencias superiores buena parte del error de cuantificación, que luego es eliminado, pero seguirá quedando un pequeño residuo de error que ha quedado dentro del ancho de banda de la señal.

Se aportan una serie de resultados representativos obtenidos al aplicar el sistema configurando diferentes parámetros para la señal "Poesía 1":

Bits de cuantificación	Factor sobremuestreo	Orden Noise-Shaping	Potencia de ruido de cuantificación
4	2	1	0 dB
4	2	2	0 dB
4	2	3	1 dB
4	4	1	-6 dB
4	4	2	-10 dB
4	4	3	-12 dB
8	2	1	-1 dB
8	2	2	-1 dB

8	2	3	0 dB
8	4	1	-7 dB
8	4	2	-11 dB
8	4	3	-15 dB
12	2	1	-2 dB
12	2	2	-2 dB
12	2	3	-2 dB
12	4	1	-8 dB
12	4	2	-12 dB
12	4	3	-15 dB

El usuario, analizando los resultados, puede comprobar que la aplicación del sistema no es trivial y no se reduce a utilizar los valores más altos posibles, es necesario apreciar la influencia de cada uno, según la situación.

La efectividad del sistema depende de forma directa del cumplimiento de la hipótesis de ruido uniforme y blanco. Para una recuantificación drástica, de 4 bits, se obtiene una reducción muy pequeña de potencia de ruido, pudiéndose, incluso, empeorar en determinados casos.

El factor de sobremuestreo juega un papel fundamental, pues con un valor tan sólo de 2, no suelen obtenerse buenos resultados; en general, como mínimo hay que emplear un factor 4 para que acción del filtro de *Noise-Shaping* sea efectiva. Se ofrece la posibilidad, además, de aplicar la técnica de Noise-Shaping eliminando la etapa de sobremuestreo. Con esto, se puede comprobar que sin ella, la aplicación de esta técnica no tiene sentido, aunque a partir de aquí pueden considerarse otros enfoques, como el *Noise-Shaping* psicoacústico.

El orden de *Noise-Shaping* es un parámetro delicado, ya que, si bien en general se reduce más ruido de cuantificación empleando un orden mayor, existe el riesgo de provocar inestabilidad. Por encima de un orden 2 (3 o mayor) no se obtiene una ventaja significativa.

La inestabilidad de la función de *Noise-Shaping* que, eventualmente, puede darse si se emplea un orden igual o inferior al número de bits de cuantificación, se advierte mediante un mensaje de alerta que aparece cuando el usuario configura los parámetros incurriendo en este riesgo. No obstante, el mensaje es sólo una advertencia, pero no impide ejecutar el algoritmo, para que el usuario analice por sí mismo este problema.

4.3 Desarrollo de la implementación en Java las aplicaciones

4.3.1 Estructura básica

Las cuatro aplicaciones web del tutorial, construidas como applets Java, se han implementado en base a una estructura de clases y métodos intentando obtener como resultado una modularización de código lo más eficiente posible; persiguiendo, además, la mayor claridad posible en cuanto a la división de tareas y procesos necesarios para la ejecución de las mismas, con el objetivo de que su comprensión sea lo más rápida y sencilla posible para futuros desarrolladores en caso de que se desee ampliarlas y mejorarlas o eventualmente, incluso, insertarlas en una aplicación superior.

Las applets constan de un código propio para lanzar cada una de ellas por separado y ejecutar su funcionalidad, y por otro, hacen uso de una serie de componentes compartidos para ejecutar ciertas funcionalidades comunes. Todo ello, en conjunto, forma el paquete "Tutorial de Digitalización de Audio".

En base a esto, las clases que forman el paquete se distinguen en dos tipos fundamentales: clases "lanzadoras" o de arranque, que son las que inician la ejecución de cada una de las cuatro applets cuando el usuario las abre al entrar en la web que las contiene, y las clases "auxiliares", que contienen una serie de métodos que realizan procesos comunes para las cuatro applets lanzadoras.

Las clases de arranque implementan el método `init()` que es el equivalente, en el mundo de las applets, al tradicional `main()` de las aplicaciones Java y otros lenguajes de programación, el cual es el método a partir del cual el programa inicia la ejecución. Estas clases contienen el código que implementa la interfaz gráfica propia de cada aplicación así como los algoritmos básicos que realizan las operaciones sobre el audio, específicas para cada applet. Reciben el nombre de la

técnica de digitalización de audio que tratan seguido del término "Applet" para indicar, con claridad, que es la clase principal de esa aplicación, es decir, la de arranque, como se ha explicado.

Las clases auxiliares, por su parte, implementan métodos estáticos que realizan procesos comunes para las cuatro aplicaciones, relacionados con la salida gráfica (representación de señales) y con operaciones esenciales de tratamiento digital de señal, tales como el filtrado, cálculo del espectro, etc. Las applets de arranque las utilizan haciendo llamadas o invocaciones a esos métodos, pasándoles una serie de datos de entrada para recibir otra serie de datos de salida u objetos que serán utilizados. Reciben un nombre genérico para denominar el tipo de procesos que llevan a cabo los métodos implementados en ellas.

Por último, se encuentran las librerías externas que han sido escogidas para realizar la funcionalidad más compleja de la aplicación y que son utilizadas por las dos tipos de clases comentadas, si bien, son las clases auxiliares las que hacen un mayor uso de ellas, pues realizan tareas generales como el cálculo de la FFT o la representación gráfica de las señales, que a su vez, son servidas a las clases de arranque tras el procesamiento correspondiente.

Las clases de arranque constan de una serie de métodos y clases internas con denominación similar en todas ellas pero que tienen una implementación propia para cada caso.

Lista de clases

- Clases de inicio o arranque
 - "QuantApplet"
 - "OversamplingApplet"
 - "DitherApplet"
 - "NoiseShapingApplet"
- Clases auxiliares
 - "Graficas"
 - "MetodosAuxiliares"
 - "GeneradorAleatorio"
 - "Conversor"
 - "Sonido"
 - "ArchivoAudio"
- Clases externas
 - "FFT" y "Complex"
 - "Librería JFreeChart"

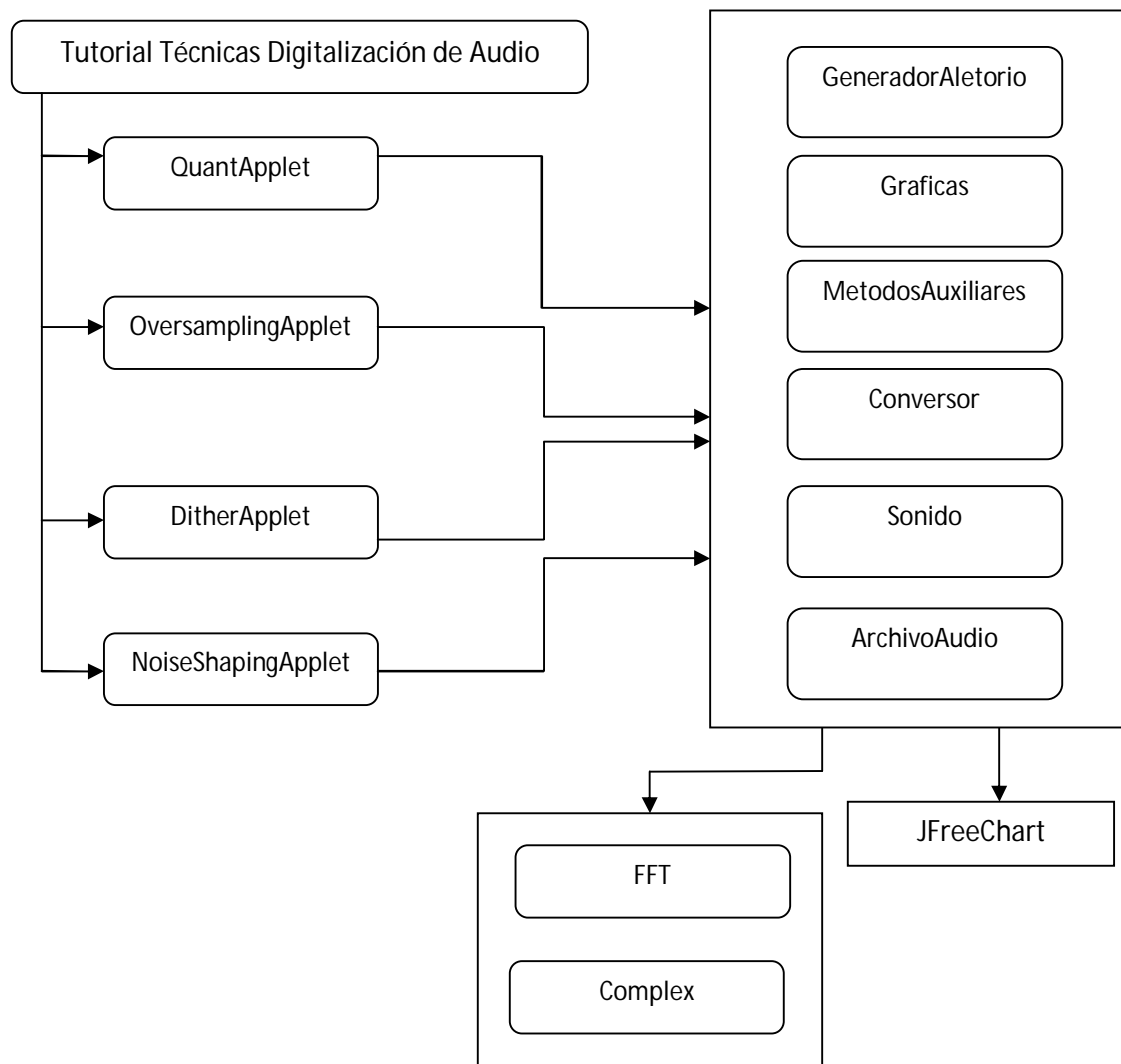


Figura 4-36 Diagrama de organización de clases

4.3.2 Clases lanzadoras o de arranque

Existen cuatro clases de arranque correspondientes a cada una de las cuatro áreas de estudio en las que se divide el tutorial. Como se ha explicado, estas clases están modularizadas en métodos y clases internas, que implementan, a diferentes niveles, procesos o subprocesos de la ejecución de la aplicación

Los métodos que poseen las clases lanzadoras son análogos, pues todas ellas siguen un esquema de ejecución similar, pero en cada una tendrán una implementación particular, correspondiente a la aplicación de la que forman parte.

4.3.2.1 *Proceso básico de ejecución*

La ejecución del programa sigue, en las cuatro applets, estas líneas básicas:

1. Creación de la interfaz gráfica (panel principal)
2. Establecimiento de los oyentes que responden a los botones de la interfaz creados
3. Estado de espera a recibir órdenes del usuario
4. Respuesta a los eventos generados por el usuario.

Tipos básicos de eventos generados por el usuario

Cargar una señal de audio

Crear el cuadro de diálogo de selección y esperar al que usuario escoja un fichero. Una vez seleccionado uno, ejecución los procesos relacionados con la extracción y conversión del archivo WAV. Resetear variables de estado y posibles señales almacenadas en ejecuciones anteriores.

El proceso de extracción se realiza, concretamente, en dos pasos esenciales:

1. Extraer las muestras que forman el archivo WAV (tipo byte)
2. Convertir las muestras tipo byte en muestras tipo float

Reproducción de señal de audio

En respuesta a la pulsación de alguno de los botones "Play", tomar la señal de audio correspondiente y convertirla a un flujo que absorberá el proceso encargado de la reproducción de sonido.

La reproducción de la señal de audio se realiza, concretamente, en dos pasos esenciales:

1. Convertir las muestras tipo float de la señal de audio a muestras tipo byte
2. Introducir las en una línea de reproducción de sonido del sistema, mediante JavaSound

Aplicar técnica

En respuesta al evento de usuario provocado por la pulsación de un botón "Aplicar técnica X", se toma la señal de audio original almacenada y se introduce en el proceso que implementa el algoritmo correspondiente, el cual devolverá la señal a la salida del sistema mas una serie de capturas de ésta en puntos intermedios de interés.

Con el resultado obtenido, se llevan a cabo tres acciones secuencialmente:

- Reproducir la señal a la salida del sistema siguiendo los pasos descritos anteriormente en "Reproducir señal de audio"
- Realizar una representación visual de las señales de interés generadas, insertándolas en el panel de gráficas como nuevas pestañas del mismo.
- Actualizar los valores numéricos de reducción de ruido de cuantificación.

En el caso de que se aplique cuantificación, se realizará un reseteo general de las variables y señales almacenadas, reiniciando la interfaz gráfica para iniciar un nuevo caso de estudio. Antes de insertar las pestañas correspondientes, se crea un panel de gráficas.

Eventos sobre el panel de gráficas

Con un panel de gráficas presente, el usuario puede realizar las siguientes acciones:

- Desplazar la señal en el tiempo hacia la izquierda o la derecha un período temporal de medio segundo. En cada ejecución, se selecciona el nuevo fragmento de la señal de audio y se solicita un objeto "ChartPanel" (ver más adelante) a la clase "Gráficas", sustituyéndolo por el anterior.
- Desplazar el slider de enventanado. Cuando el slider cambia de posición, se selecciona la ventana de señal correspondiente y se calcula su FFT, el resultado obtenido (espectro) se representa visualmente de forma inmediata pidiendo un objeto gráfico a "Gráficas"

4.3.2.2 *Métodos y clases internas*

Los métodos y clases internas tienen la misma denominación en todas las clases, pero en cada una de ellas tienen una implementación particular, si bien, en general, el núcleo es el mismo pero modifican variables de control propias.

4.3.2.2.1 Métodos

`init`

Método que arranca la ejecución del applet invocando el método

`interfazGrafica`

Crea y sitúa los elementos gráficos de la interfaz del panel principal de la aplicación, asignando los oyentes que recogerán los eventos de los botones correspondientes.

La interfaz gráfica consta de dos áreas esenciales: la de fichero de audio y la de trabajo. La de trabajo, a su vez, está dividida en tres zonas: configuración de parámetros de aplicación de la técnica, panel de gráfica y resultados numéricos.

`extracción`

Abre el fichero de audio seleccionado por el usuario creando un objeto de la clase "ArchivoAudio" que contendrá el contenido de éste con muestras de tipo byte y los datos correspondientes.

`transformación`

Convierte una señal de audio formada por muestras tipo byte (provenientes de un fichero WAV) a muestras tipo float con rango -1/1, haciendo uso del método "transformacion" de la clase estática "Conversor, al cual le envía los canales que forman la señal (uno en caso de mono, dos en caso de estéreo)

`transformacionInversa`

Convierte una señal de audio formada por muestras tipo float, provenientes de la salida de un sistema de la aplicación, a muestras tipo byte haciendo uso de la clase "Conversor", para ser reproducida.

`almacenamiento`

Se guarda la señal de audio extraída de un fichero WAV en las variables globales que manejará la aplicación. En caso de que se trate de una señal estéreo, divide el array de bytes que contiene los canales izquierdo y derecho multiplexados, en dos arrays de bytes.

`reproducir`

Reproduce la señal de audio correspondiente haciendo uso del método `repAudio` de la clase estática `Sonido`.

`cuantificador`

Introduce la señal original en el sistema de recuantificación, almacenando el resultado (señal cuantificada) en la variable `yA` y calcula el error provocado, que almacenará en la variable `eA`.

Se “resetea” el estado de la aplicación en caso de que se hayan producido ejecuciones previas de la cuantificación o la técnica tratada por el applet.

`crearVentanaGrafica`

Crea el panel de solapas de las gráficas, insertando la primera de ellas: la gráfica de la señal original.

`ampliarVentanaGrafica`

Creación e inserción en el panel de gráficas de las solapas restantes cuando se aplica la técnica correspondiente. Las solapas están vacías inicialmente y su contenido es representado cuando son pulsadas.

`actualizarInfoAudio`

Actualización de los datos de información del fichero de audio cargado por el usuario (nombre, duración, frecuencia de muestreo y bits de cuantificación)

`interfazAbrir`

Creación del cuadro de diálogo de selección de fichero de audio

`resultadosNumericos`

Actualización de los resultados numéricos (potencia del error de cuantificación) tras ejecutarse la aplicación de la técnica.

4.3.2.2.2 Clases internas

`OyenteAbrir`

Responde a la pulsación del botón "Abrir" ejecutando el método `interfazAbrir`.

`OyenteMenuAbrir`

Responde a la pulsación de cualquiera de los tres botones de carga de fichero de audio que contiene el cuadro de diálogo "Cargar fichero de audio".

`OyenteRep`

Responde a la pulsación de cualquiera de los botones "Play" ofrecidos por la interfaz, ejecutando la reproducción de la señal correspondiente al botón pulsado. El proceso se divide en dos pasos básicos:

1. Transformación de la señal con muestras tipo float en muestras tipo byte, mediante una invocación del método "transformaciónInversa" que recibe la señal que corresponda.
2. Reproducción de la señal mediante una llamada al método reproducir.

`OyenteCuantificacion`

Responde a la pulsación del botón "Aplicar cuantificación" invocando el método `cuantificación`, el cual aplica el proceso.

Una vez obtenida la señal cuantificada, se reproduce mediante la invocación de los métodos "transformaciónInversa" y "reproducir".

`OyenteValoresOptimos`

Responde a la pulsación del botón "Configurar valores óptimos" en las applets Sobremuestreo, Dither y "Noise – Shaping", actualizando los parámetros de configuración de las técnicas en función del número de bits de cuantificación seleccionado.

OyenteSobremuestreo, OyenteDither, OyenteNoiseShaping

Respuesta a la pulsación de los botones "Aplicar sobremuestreo / cuantificación", "Aplicar Dither / cuantificación" y "Aplicar Noise - Shaping / cuantificación", respectivamente, ejecutando la aplicación del algoritmo correspondiente en cada caso.

Los resultados, tanto la salida del sistema como la señal y el error en diferentes puntos de interés de éste, se almacenan en variables globales que serán utilizadas por los módulos correspondientes de reproducción, representación gráfica, etc.

OyenteBarra

Respuesta al desplazamiento del slider de enventanado sobre una gráfica de tiempo. Este desplazamiento supone centrar la ventana de FFT en un nuevo instante temporal de la señal al cual le corresponderá una nueva densidad espectral de potencia.

Se obtiene un objeto "JFreeChart" que sustituirá a la anterior gráfica DEP, mediante una llamada al método panelGrafica o panelGraficaMultiple, el cual, recibirá por parámetro la nueva posición del slider.

OyenteDesplazamiento

Respuesta a los botones de desplazamiento "<" y ">" situados en la parte superior de la gráfica de tiempo.

Cuando se produce un desplazamiento, se muestra un nuevo fragmento de la señal de audio de medio segundo, centrado una cuarta parte de medio segundo de la señal de audio.

OyenteTabs

Respuesta a la pulsación de cualquiera de las solapas contenidas en el panel de gráficas.

Se crean dos objetos "JFreeChart", correspondientes a la gráfica en tiempo y en espectro, y los elementos de interfaz necesarios (slider, botones de desplazamiento, etiquetas, etc)

4.3.3 Clases auxiliares

4.3.3.1 *ArchivoAudio*

Clase que tiene la función de extraer y guardar el contenido de un archivo de audio WAV y la información referente a él.

Las applets de arranque crean un objeto "ArchivoAudio" cuando el usuario selecciona un archivo para cargarlo en la aplicación y trabajar con él. Dicho objeto tiene ámbito global dentro de las clases de arranque y por tanto, pueden acceder a él cualquiera de los métodos que componen el applet.

El objeto de la clase "ArchivoAudio" se crea pasándole, como único parámetro, la dirección en la cual se encuentra el archivo en cuestión. Esta dirección debe tener formato de URL y puede tratarse de una dirección de Internet o una ruta local de un fichero en disco.

Cuando se crea el objeto, en el constructor, se ejecutan los procesos necesarios para extraer del archivo las muestras de audio que lo forman (en formato byte) así como la información sobre el archivo. Por tanto, no es necesario ejecutar ningún método para extraer las muestras una vez creado el objeto "ArchivoAudio".

Para extraer los datos mencionados (contenido e información) se hace uso de varias clases del paquete Java Sound que hacen posible la implementación de esta tarea:

- `AudioInputStream`
- `AudioFileFormat`
- `AudioFormat`
- `AudioSystem`

El proceso es el siguiente:

1. Obtención del flujo de entrada de audio y formato

La extracción del fichero de audio comienza con la obtención un flujo de entrada de audio al sistema, a través la clase `AudioSystem`, que proporciona este elemento mediante el método estático `getAudioInputStream`. Este método recibe como parámetro la dirección en la que se encuentra el archivo.

El flujo de entrada (`AudioInputStream`) es el punto de partida para operar con el archivo de audio. A partir de él se podrán extraer las muestras que lo componen.

La información relativa al archivo se obtiene a partir del objeto `AudioFileFormat`, con el método `getAudioFileFormat` de la clase `AudioSystem`, el cual recibe, al igual que sucedía con el flujo de entrada, la dirección en la que se encuentra el archivo. Luego, será necesario crear un objeto de la clase `AudioFormat`, que es el que, efectivamente, contendrá los datos del archivo, a partir del método `getFormat` del objeto `AudioFileFormat` que se ha creado previamente.

Por tanto, en resumen, se puede decir que `AudioSystem` es la clase “madre” que genera los elementos básicos para el procesado de audio en Java, los cuales son: flujo de entrada (`AudioInputStream`) y datos de formato de archivo (`AudioFileFormat`)

2. Extracción de las muestras

Las muestras del archivo de audio se extraerán secuencialmente leyendo los bytes que lo componen, a través de bloques de tamaño configurable y quedarán almacenadas en el array de bytes “`audioBytes`”.

En primer lugar, se obtiene el valor de ciertos datos del archivo, necesarios para llevar a cabo el proceso de extracción. Tales datos son el número de frames que componen el archivo y el número de bytes contenidos en cada frame. Los “frames” son fragmentos cortos, de cierto número de bytes fijo, en los que `JavaSound` divide el archivo de audio para poder manejarlo. La multiplicación del

número de frames por el número de bytes por frame, por tanto, indica la longitud total, en bytes, del fichero de audio procesado.

Una vez conocida la estructura interna del archivo, se procede a leerlo secuencialmente.

Se crea una variable de intercambio denominada "byteLeído", que consiste en un array de pequeño tamaño que almacenará el bloque leído en cada paso y otra variable, denominada "audioBytes", que será en la que se guarden los bloques que sucesivamente se extraigan.

Se ejecuta dentro de un bucle el método "read" del flujo de audio creado anteriormente, tantas veces como bloques compongan la señal. Este método recibe por parámetro el array de intercambio "byteLeído" en el cual volcará los bytes que componen el bloque. La posición de lectura avanza automáticamente en cada ejecución del método read, situándose en el siguiente bloque una vez termina.

Por ejemplo, si el tamaño del bloque son L bytes, y se han leído x bloques, el byte a partir del cual leerá en la siguiente ejecución del método será $L \cdot x$.

Es importante destacar la diferencia entre el bloque y los frames. Los frames están relacionados con la estructura interna del `AudioInputStream`, la cual, carece de interés para la aplicación. El único objetivo por el que se obtiene el número de frames y su tamaño en bytes es conocer el número de bytes total que componen el archivo y por tanto, saber hasta dónde hay que leer.

El bloque, en cambio, es una estructura de almacenamiento creada por el programa. Hace uso del `AudioInputStream` para extraer secuencialmente, en trozos pequeños, las muestras que componen el archivo.

Este algoritmo, que se puede considerar de cierta complejidad, es necesario implementarlo de esta manera para una ejecución exitosa de la lectura del archivo. Podría caerse en la tentación de pasar directamente al método read un array de tamaño el número total de bytes que forman el archivo, para así, ahorrarse el bloque de intercambio y el bucle, pues volcaría directamente el archivo de audio entero. Si esto se hace así, por cuestiones de carga computacional, el método falla y no se obtiene correctamente el flujo creado.

La elección del tamaño del bloque es, por tanto, una decisión importante para lograr un correcto funcionamiento de la aplicación. Se ha constatado, además, la influencia de este valor dependiendo del escenario de ejecución de la aplicación. Si el archivo de audio está almacenado en el disco duro local, el algoritmo puede funcionar con tamaños relativamente grandes de bloque. En cambio, si el archivo está colgado en Internet, es necesario utilizar bloques pequeños.

El tamaño elegido es 128 bytes (el hecho de que este valor sea potencia de dos es casual, no es un requisito). Con este valor, se ha constatado que la lectura se realiza correctamente en todos los escenarios de uso considerados de la aplicación.

Por último, indicar que en el caso de archivos mono el array de bytes obtenido puede ser empleado tal cual para reproducirlo por una línea. En el caso de archivos estéreo, el array contiene multiplexados los canales izquierdo y derecho, de manera simple: se alternarán sucesivamente un byte del canal L y un byte del canal R.

3. Extracción de la información

Se almacenan en las correspondientes variables del objeto `ArchivoAudio` ciertos datos del archivo que pueden ser de interés para las applets que lo utilizan, los cuales son:

- Tipo de codificación
- Frecuencia de muestreo
- Número de bits de cuantificación
- Número de canales (mono / estéreo)
- Tamaño de "frame"
- Tasa de "frame"
- Tipo de codificación binaria (bigEndian / littleEndian)

4.3.3.2 *Conversor*

La clase `Conversor` es la encargada de transformar las muestras de audio de tipo byte en muestras tipo float con fondo de escala +1 / -1, que pueden manejar los algoritmos de procesamiento de audio implementados por la aplicación. Alternativamente, realiza el proceso inverso, paso de float a byte, para que una vez

la señal de audio ha sido procesada por el sistema, sea introducida en un `AudioInputStream` y sea reproducida.

Consta de dos métodos estáticos que realizan las dos operaciones mencionadas.

transformación

El método `transformación` devuelve array de muestras tipo `float` a partir de un array de muestras tipo `byte` que recibe por parámetro. Puede recibir por parámetro el array de audio mono o los dos canales de audio estéreo y devolverá lo indicado por la variable retorno (señal mono, canal izquierdo de señal estéreo o canal derecho de señal estéreo).

Para ello, en primer lugar, discrimina entre una serie de casos dependiendo de las características del archivo de audio: número de canales (mono o estéreo), número de bits de cuantificación (8 o 16) y tipo de codificación binaria (tipo `Little Endian` o `Big Endian` en el caso de 16 bits).

En el caso de 16 bits de cuantificación, la operación consiste en componer una muestra tipo `short` (16 bits) a partir de los dos bytes que la forman, mediante operaciones lógicas y de desplazamiento. El byte "High" se desplaza 8 posiciones a la izquierda y el derecho se mantiene igual. Luego se aplica el operador `OR` para obtener la muestra de 16 bits.

```
b1s=(short)(bHigh<<8);
b2s=(short)(bLow&0xFF);
samp = (short)(b1s | b2s);
```

El formato `BigEndian` o `Little Endian` marca la posición en la que vienen insertados el byte "High" y el byte "Low" para cada muestra. En el caso `BigEndian` el primer byte es el "High" mientras que para `LittleEndian`, primero se inserta el byte "Low" para cada muestra. Se trata, por tanto, de una mera cuestión de implementación, pero esta opción no influye en la calidad del audio.

Después, esta muestra, que tiene una precisión de 2^{16} valores, es decir, oscila entre -32767 y +32766, se divide por 32767 y se hace un casting a `float`, para obtener, finalmente, la muestra deseada tipo `float`.

Para audio de 8 bits de cuantificación, el proceso es similar pero saltando el paso de combinación de los bytes "High" y "Low" ya que cada muestra se compone de un único byte.

`transformacionInversa`

Se aplica el mismo proceso descrito para el método transformación, pero de manera inversa. En primer lugar, se multiplica cada muestra por 32767 y después, para el caso de 16 bits, se crean los dos bytes que se insertarán en el flujo mediante las operaciones lógicas inversas.

4.3.3.3 Sonido

La clase Sonido es la encargada de reproducir por los altavoces del PC en el que se ejecuta la aplicación una señal de audio, recibiendo como entrada un array de bytes.

Al igual que la clase ArchivoAudio, hace uso del paquete JavaSound para realizar esta tarea de su funcionalidad. El proceso se realiza llevando a cabo los siguientes pasos:

1. Creación de la señal de audio

Se crea una señal de audio encapsulada en el objeto `AudioInputStream` pasándole el array de bytes y los datos de formato y codificación que contiene el objeto `ArchivoAudio`

2. Creación de una línea de reproducción

Se crea una línea de reproducción que es la vía de comunicación entre JavaSound y la tarjeta de sonido. La línea, que la implementa un objeto de la clase `SourceDataLine`, se obtiene con el método `getLine` de la clase `AudioSystem`. Luego, se establece el formato y tipo de codificación de la señal de audio que recibirá la línea con el método `open`.

3. Reproducción de audio

La reproducción de audio se realiza pasando de manera secuencial a línea de reproducción bloques de muestras.

En primer lugar, se vuelca un bloque de muestras en el array de bytes tempBuffer con el método read de ArrayInputStream. Luego, este bloque, se inserta en la línea de reproducción con el método write de SourceDataLine. A partir de aquí, las muestras son enviadas a la tarjeta de sonido y reproducidas por los altavoces.

4.3.3.4 MétodosAuxiliares

La clase MétodosAuxiliares implementa una serie métodos estáticos relacionados con el procesado digital de señales de audio que son utilizados por las clases de arranque para aplicar sus algoritmos de digitalización de audio.

`cuantificarMuestra`

Cuantifica una muestra de audio x con un escalón de cuantificación q , aplicando redondeo con el método round de la clase Math. Tras ello, rectifica la muestra si tiene un valor mayor que 1 o menor que -1 para evitar el desbordamiento del fondo de escala.

`interpolar`

Realiza una interpolación de un array de muestras de audio de longitud L , cuyo resultado un array de longitud $L \cdot \text{factor}$.

Para ello, se da valor a las nuevas muestras, que son introducidas entre las originales, aplicando un filtro paso bajo de primer orden. Es decir, se suaviza la forma de onda calculando una recta entre dos muestras originales consecutivas con la cual se dará valor a las muestras artificiales introducidas entre ambas.

`diezmar`

Realiza un diezmando de un array de muestras de audio de longitud L , cuyo resultado es un array de muestras de longitud L/factor .

Para ello, selecciona muestras del array original en pasos $x + \text{factor}$.

`filtrar`

Aplica un filtro paso bajo sobre la señal de audio que recibe con frecuencia de corte relativa f (esto es, $f_{\text{corte}} = f \cdot f_s$, es decir, si la señal está muestreada a 11025 Hz, y se aplica $f=0.5$, la frecuencia de corte será 5512.5 Hz).

Para ello, en primer lugar, se definen los coeficientes del filtro paso bajo a aplicar, y luego, en un bucle, se obtienen las muestras filtradas, que se calculan como la suma del valor de cada muestra y sus vecinas ponderadas con el coeficiente que corresponda.

`calcularPotenciaDEP`

Realiza el cálculo de la potencia de señal devolviéndolo en decibelios. Para ello, aplica el sumatorio de las muestras elevadas al cuadrado y divididas por el número total de muestras.

`enventanar`

Aplica una ventana centrada en un punto de la señal. En primer lugar, selecciona las muestras de audio que formarán parte de ella, lo cual, es equivalente a aplicar una ventana uniforme. Luego, aplica la ventana sobre dichas muestras empleando la fórmula de Hamming.

El conjunto de muestras devuelto debe tener un tamaño potencia de dos ya que es un requisito del método FFT, el cual lo tomará posteriormente para calcular su espectro.

`obtenerFactorCuantificacion`

Recibe el número de bits seleccionados por el usuario para realizar la cuantificación de la señal y en base a él, calcula el tamaño del escalón de cuantificación, que posteriormente, será utilizado para cuantificar las muestras.

`factorPonderacionVentanaHanning`

Calcula el factor de ponderación de la ventana Hanning, valor que será utilizado para aplicar la fórmula de enventanado Hanning.

4.3.3.5 Gráficas

Esta clase consta tres métodos estáticos que implementan la creación de las gráficas de señal en tiempo y frecuencia utilizadas por las cuatro applets de arranque que se insertan en el panel de solapas. Los métodos devuelven un objeto JFreeChart a partir de la señal (o señales) que se les pase y ciertos parámetros de control.

Las clases de arranque crearán un "ChartPanel" a partir del objeto JFreeChart que devuelven estos métodos, el cual rellenará la solapa correspondiente a la gráfica. El objeto "ChartPanel" implementa las utilidades que JFreeChart ofrece aparejadas a las gráficas, tales como el zoom, guardado en un archivo png o gif, aspecto visual, etc. El objeto "ChartPanel" no es más que una extensión del objeto "JPanel" del swing de Java.

Puesto que algunos de los parámetros de "ChartPanel" deben de configurarse de forma específica en cada applet para cada gráfica, la creación del objeto ChartPanel se realiza en ellas, en vez de que la clase Graficas lo devuelva directamente.

Los métodos de la clase Graficas son:

- panelGrafica
- panelGraficaMultiple
- panelGraficaPromedio

El método panelGrafica genera una gráfica de una señal única, el método panelGraficaMultiple genera una gráfica de dos señales superpuestas con la finalidad de ser comparadas.

Estos dos métodos pueden devolver una gráfica en tiempo o una gráfica de densidad espectral de potencia según se indique en el parámetro "fft" ("true" si se desea obtener el espectro, "false" si se desea obtener la forma de onda en tiempo).

El método panelGraficaPromedio obtiene el espectro la señal implementando la técnica de Welch. Consiste, esencialmente, en calcular la densidad espectral de potencia de una serie de ventanas de señal (periodogramas) de forma secuencial, las cuales serán luego promediadas con el objetivo de obtener

la DEP media de toda la señal. Su utilidad se encuentra, sobretodo, en el análisis del espectro del ruido de cuantificación, el cual, en instantes concretos, puede tomar una forma arbitraria, más o menos cercana a la hipótesis de ruido uniforme y blanco, promediando se puede comprobar que, efectivamente, tiende a adoptar el comportamiento. El método recibe dos señales de entrada, pues será utilizado en todas las applets para realizar un análisis comparativo del espectro de ruido de cuantificación de la señal de entrada y la de salida.

4.3.3.5.1 Gráficas en tiempo

En caso de que se solicite la gráfica en tiempo, los métodos de esta clase construirán ésta seleccionando un fragmento de medio segundo (5512 muestras en el caso de audio muestreado a 11025 Hz) a partir del instante temporal indicado por el parámetro "puntoReferencia". El desplazamiento del usuario a través de fragmentos consecutivos de señal, por tanto, es transparente para estos métodos, es decir, este proceso lo gestionarán las clases de arranque, que guardarán el estado de las diferentes gráficas en variables internas, solicitando, en cada actualización (cuando el usuario pulsa los botones de desplazamiento) el fragmento que corresponda.

La elección de medio segundo como longitud del fragmento de audio que será mostrado en la gráfica de forma de onda se ha realizado en base al rendimiento observado de JFreeChart. La creación de gráficas con un número elevado de muestras es costosa computacionalmente y en ordenadores de gama media, de cierta antigüedad, se podrían producir ralentizaciones en el uso del applet. Para mejorar la experiencia de usuario, se ha buscado un valor de compromiso, el cual, ha sido, en efecto, medio segundo, 5512 muestras para señales con frecuencia de muestreo 11025 Hz. Con dicho valor, las gráficas se crean de forma casi instantánea, en un tiempo inapreciable para el usuario y el período que muestran, medio segundo, es suficiente para navegar con comodidad por la señal.

Para mejorar aun más el tiempo de creación de gráficas se podría haber tomado la opción de diezmar la señal, puesto que realmente, no es necesario pintarlas todas debido a la integración espacial que realiza el ojo (por ejemplo, pintando 1000 muestras, se observa una gráfica con el mismo aspecto que con

5512 muestras). Sin embargo, esto restaría profundidad al zoom, del que se requiere que llegue al máximo posible, es decir, poder llegar a observar cada una de las muestras individualmente y así, especialmente para el applet de *dither*, observar los escalones de cuantificación. Por otro lado, diezmando, se correría el riesgo de obtener unas representaciones muy deficientes, distorsionadas, de algunas señales particulares, por ejemplo, tonos de determinada frecuencia o señales con transitorios especiales.

Por estos motivos, las señales tienen que ser representadas con todas las muestras, lo cual supone un coste computacional importante, obstáculo que se salva renunciando a realizar una representación completa de la señal, de principio a fin, en una sola vista, en cambio, se representa un fragmento de ella poniendo a disposición del usuario botones de desplazamiento con los que podrá navegar hacia delante o hacia atrás.

Hay que tener en cuenta que para señales con frecuencia de muestreo mayor (44100 Hz típicos de CD) o señales sobremuestreadas, el número de muestras será superior para medio segundo, por tanto, aumentará de forma proporcional el tiempo de espera en la creación de las gráficas, si bien, se ha comprobado que es un retardo razonable y prácticamente inapreciable.

El proceso de creación de una gráfica sigue estos pasos:

1. Selección del fragmento de audio

Se crea un array (variable `arrayFragmento`) donde se almacenará el fragmento de audio, a partir de la selección de la parte de señal comprendida entre el punto de referencia indicado por la clase que solicita la gráfica (variable `puntoReferencia`) y el número de muestras que forman medio segundo de señal, en función de la frecuencia de muestreo y el factor de sobremuestreo.

2. Inserción de las muestras del fragmento en el objeto `XYSeries`

Las muestras del array creado (`arrayFragmento`) se insertan en un objeto `XYSeries`, el cual, es la estructura esencial que la librería `JFreeChart` utiliza para definir las muestras con las que se construirán las gráficas que se desean obtener.

Cada muestra que se añade al objeto XYSeries mediante el método add, tiene dos elementos (que son los parámetros que recibe dicho método) ,el valor de la muestra en el eje X y el valor en el eje Y. En este caso, instante de tiempo y valor de amplitud de la señal para él, respectivamente

El proceso es análogo para el caso de panelGraficaMultiple, pero aplicándolo, de forma paralela, a las dos señales que recibe. Los dos arrays creados se insertan en dos objetos XYSeries distintos, que serán añadidos al mismo objeto JFreeChart, el cual, los mostrará superpuestos.

3. Creación del objeto JFreeChart y ajustes

Se crea un objeto XYSeriesCollection, el cual contendrá el conjunto de series que aparecerán en la gráfica, una señal en el caso del método "panelGrafica" y dos señales en el caso del método "panelGraficaMultiple". Luego se crea el objeto JFreeChart a partir de la colección de señales generada antes y otra serie de parámetros de control.

Finalmente se realiza, opcionalmente, un ajuste del número de marcas en el eje de ordenadas (Y) y su disposición, para obtener una representación correcta y evitar solapamientos; asimismo, se transforma el formato numérico de las marcas para mostrarlas con notación científica. La aplicación o no de esta parte se indica mediante la variable "max".

4.3.3.5.2 Gráfica de espectro

En el caso de que se solicite una gráfica de espectro se selecciona el fragmento de señal sobre el cual se aplicará, se envientana, se aplica la FFT y con ésta, se calcula Densidad Espectral de Potencia.

El proceso, detallado, se realiza así:

1. Selección de un fragmento de medio segundo

Se selecciona el fragmento de medio segundo de la señal de audio correspondiente al que se muestra en pantalla y se almacena en la variable "arraySeccionado".

2. Enventanado

Se aplica una ventana, tipo Hanning, sobre el fragmento seleccionado, mediante el método "enventanar" de la clase "MetodosAuxiliares" (para más detalles sobre este método, consultar documentación) y se guardan en la variable "arrayEnventanado".

3. Conversión a formato Complex

Los valores tipo float de las muestras del fragmento de audio enventanado se convierten en valores tipo Complex, clase externa utilizada para la representación en forma de número complejo (en este caso, solo con parte real) y que la clase FFT, posteriormente, tomará para calcular el espectro. Las muestras, transformadas a tipo Complex, se guardan en el array "c1".

4. Cálculo de la transformada de Fourier

Se calcula el espectro del fragmento enventanado mediante el método estático "fft" de la clase externa "FFT", el cual recibe el array de muestras tipo Complex previamente construido y devuelve otro array de muestras Complex.

5. Cálculo del factor de ponderación

Para el cálculo de la densidad espectral de potencia, es necesario obtener antes un factor que se empleará en la fórmula de cálculo de la DEP (el cual divide a cada una de las muestras de FFT, ver capítulo de teoría). Este factor se calcula en función de la ventana aplicada y el número de muestras. Para obtenerlo se emplea el método "factorPonderacionVentanaHanning" de la clase "Metodos Auxiliares".

6. Cálculo de la densidad espectral de potencia

Se toma el array de muestras de la FFT y con el factor de ponderación calculado antes, se obtiene la densidad espectral de potencia. Esto se realiza elevando las muestras de FFT al cuadrado y multiplicándolas por el factor de ponderación (ver teoría para mas detalles).

7. Inserción de las muestras de Densidad Espectral en el objeto XYSeries

Se añaden las muestras de densidad espectral de potencia al objeto XYSeries. Para ello se pasan por parámetro al método add los valores de las componentes en el eje X y en el eje Y de la muestra de FFT, que son, frecuencia y magnitud, respectivamente.

8. Creación del objeto JFreeChart y ajustes

Se crea, finalmente, el objeto JFreeChart siguiendo los mismos pasos descritos en "Gráficas en tiempo" y se realizan los ajustes pertinentes sobre las marcas del eje de ordenadas.

4.3.3.6 Generador Aleatorio

Consta de tres métodos estáticos que devuelven un array de números aleatorios del tamaño especificado, los cuales, serán empleados como fuente de *dither*. Cada método genera los números aleatorios de acuerdo a una determinada función densidad de probabilidad y ciertos parámetros configurables de ésta.

- arrayUniformeFloat
- arrayGaussFloat
- arrayTriangularFloa

V

CONCLUSIONES Y TRABAJO FUTURO

5 Conclusiones y trabajo futuro

5.1 Conclusiones

El proyecto que se ha llevado a cabo tiene un carácter, bajo la perspectiva del I+D+i, mucho más cercano al concepto de desarrollo y de innovación, que al de investigación, esto es, aprovechar el conocimiento científico existente para crear una solución de ingeniería, y no tanto, crear más conocimiento científico en el ámbito tratado. Por este motivo, las conclusiones que se pueden extraer, se orientan más bien al grado de éxito en la consecución de los objetivos y en la experiencia que se ha adquirido a la hora de abordar el problema, y como puede servir ésta para que en futuros proyectos, éstos se realicen con más eficacia.

El proyecto se ha realizado bajo la premisa de crear una plataforma innovadora, de apoyo al estudio de una materia de ingeniería, en concreto, las técnicas de digitalización de audio. Se ha partido de una idea esencial, que es combinar la presentación de la teoría junto con ejercicios prácticos del modo más natural posible, facilitando el aprendizaje. El grado de éxito de este objetivo solo se podrá verificar *in situ*, cuando los usuarios, en principio alumnos universitarios, aunque es válida cualquier persona interesada en la materia, utilicen la aplicación y se compruebe su valor pedagógico. A falta de este paso, las presentaciones hechas a diversos sujetos, de muy diverso nivel académico e intereses, han sido satisfactorias y destacan por la originalidad que supone implementar una herramienta que complementa la teoría expuesta en una clase universitaria. La facilidad de uso y la claridad con la que se exponen los contenidos son puntos fuertes; por otro lado, el funcionamiento de las applets cumple los requisitos mínimos exigibles, aunque siempre, como todo producto software, es susceptible de ser mejorado y ampliado.

El resultado final, entrando en detalle, tiene varios componentes que pueden ser analizados, la teoría por un lado y los ejercicios prácticos, es decir, los applets, por otro.

La exposición, en el tutorial, de la teoría, se ha realizado buscando la mayor sencillez y claridad posible, de la manera más cercana al alumno. Se ha encontrado

que las explicaciones existentes en las fuentes bibliográficas más importantes eran breves y áridas, lo que supone una barrera de entrada para el estudio de las técnicas de digitalización de audio, tan importantes para un ingeniero de sonido. Además, por el hecho de no ser una de las materias de estudio más atractivas en el campo del audio, en contraste, por ejemplo, con la ecualización de recintos o la creación de efectos de reverberación, su estudio, muchas veces, no recibe toda la atención que merece. Sin perder rigurosidad en las demostraciones matemáticas y la explicación de los algoritmos, se ha intentado hacer una presentación atractiva, que despierte el interés por la materia, necesario para que se prosiga en la investigación de campos como el *Noise-Shaping* psicoacústico.

Este razonamiento se puede extrapolar al estudio de cualquier rama de la ingeniería en general, y no sólo quedarse en el campo del tratamiento digital de audio. Es necesario hacer un esfuerzo por crear material didáctico en el que se muestre los conceptos teóricos básicos de un modo coloquial, para que más tarde, el estudiante, si lo necesita, acceda a bibliografía especializada en la que ya sí, se presenten los contenidos de una manera más formal y exhaustiva. El objetivo de la ingeniería, al fin y al cabo, es crear soluciones tecnológicas que aumenten el bienestar de la gente, por tanto, su estudio debe facilitarse al máximo y hacerlo lo más accesible que sea pueda. Esta idea, además, adquiere especial relevancia en el momento actual, con la implantación del Espacio Europeo de Educación Superior. Proseguir en el desarrollo de proyectos de este tipo es altamente recomendable y supondría un revulsivo en la pedagogía universitaria. Respecto a su vertiente en la tele-educación, también, es una buena manera de profundizar en el concepto de *e-learning* como recurso electrónico de apoyo al aprendizaje.

En cuanto al desarrollo de las aplicaciones web que implementan los ejercicios prácticos, su resultado, como ya se ha dicho, ha sido satisfactorio y cumple los objetivos mínimos marcados al comienzo del proyecto, aunque no obstante, como es natural, son mejorables. El estudio del estado del arte se ha comprobado que es un factor clave a la hora de abordar cualquier solución informática, especialmente, en el ámbito software libre. El tiempo empleado en buscar librerías que ayuden a la implementación del objetivo, o la búsqueda de soluciones ya existentes previamente, para ser tomadas como punto de referencia,

es esencial y nunca un desperdicio. Es más, hay que tener presente que en la ingeniería, los avances son muy rápidos, y esta tarea, nunca debe abandonarse en todo el tiempo dedicado a un proyecto, adaptándose a las circunstancias si es necesario. Por otro lado, dado que el producto desarrollado es elemental y ha podido ser codificado por un solo programador, no se ha estimado oportuno dedicar tiempo a plantear un proceso de desarrollo formal. No obstante, se puede decir, que en la práctica, se ha seguido un proceso de desarrollo incremental, a partir de unas especificaciones iniciales, se han implementado las funcionalidades prioritarias, y una vez verificado su correcto funcionamiento, se ha proseguido con las siguientes en orden de importancia. Este método se ha demostrado efectivo para un proyecto de estas características, en el que no es importante la coordinación de un equipo de programadores, pues no existe, pero sí verificar que la solución desarrollada se va ajustando a los requisitos de partida.

Respecto a la solución obtenida en sí misma, se ha logrado el objetivo de programar una aplicación que amplíe las funcionalidades de las ya existentes y que aporte una visión innovadora. Así, ya se puede comprobar de forma cuantitativa el resultado de aplicar las técnicas de digitalización de audio, y se posee una herramienta para inspeccionar de un modo más exhaustivo el espectro, haciendo comparaciones con diferentes señales superpuestas, en diferentes etapas de los sistemas. Esto ayuda notablemente a la comprensión de dichas técnicas, y hasta ahora, no existía ninguna aplicación que lo ofreciese. No obstante, para lograrlo, ha habido que pagar un precio, no se han podido incluir algunas de las funcionalidades que sí tienen las aplicaciones ya existentes, como es observar en tiempo real, de forma sincronizada, el efecto de cambiar parámetros mientras se reproduce la señal de audio, pues la utilización de una librería de gráficas, con unos requerimientos computacionales elevados, lo impedía. El resultado final, por tanto, hay que verlo como una alternativa a las aplicaciones ya existentes, un punto de vista diferente y nuevo, que las complementa, pero no como una versión mejorada que las sustituya.

5.2 Trabajo futuro

El resultado final de este proyecto puede ser mejorado o complementado en algunos puntos en los que se ha considerado más oportuno que sean tomados en el futuro para dedicarles una atención exclusiva, por su complejidad y extensión del trabajo a realizar.

- El estudio práctico de *Noise-Shaping* puede ampliarse a funciones avanzadas que implementen un modelo psicoacústico. En el tutorial se ha dedicado un capítulo a esta técnica, dando una visión general de su efecto y algunas nociones de las propuestas que han realizado diferentes grupos de investigación para implementarla, pues conllevan una algoritmia compleja. Complementar el applet de *Noise-Shaping* con esta técnica mejoraría notablemente la experiencia del usuario, pues es el único concepto del tutorial que sólo se puede presentar de forma teórica y del que no se pueden realizar ejercicios prácticos.

En particular, se recomienda tomar como punto de partida el trabajo [9] pues el más avanzado y el que llega a las conclusiones más específicas, respecto a la realización práctica del sistema, aunque, como es obvio, al tratarse éste de un tema de estudio en evolución en la actualidad, es recomendable revisar el estado del arte del mismo e incluso, cabe la posibilidad de que sea necesario reescribir el capítulo “Modelado psicoacústico” poniéndolo al día de los últimos avances.

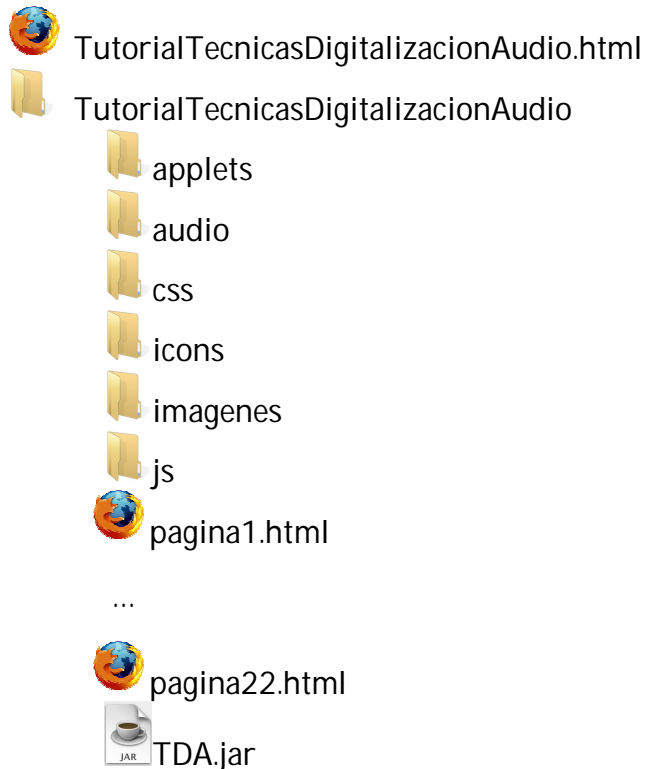
- Introducir la posibilidad de modificar los parámetros de las técnicas en tiempo real, mientras se reproduce la señal de audio, y que además, la representación del espectro se actualice de forma dinámica. Esta funcionalidad se ha desechado inicialmente pues tiene un carácter más vistoso que didáctico, no es imprescindible para el estudio de las técnicas, por lo que se ha preferido profundizar en otras opciones, ya que su coste computacional elevado, aunque su inclusión mejoraría el atractivo de las applets.
- Realizar una mejora general del funcionamiento y rendimiento de las aplicaciones, especialmente, en lo referente al cómputo de la densidad de espectral de potencia y su representación gráfica, profundizando en

sacar más partido a la librería JFreeChart e incluyendo posibles mejoras que traigan nuevas versiones de ella.

- Estudiar la problemática de la estabilidad del filtro de *Noise-Shaping* para órdenes elevados, y eventualmente, aplicar algún método para solucionar o mitigar esta cuestión y así, poder incluir órdenes superiores [22].
- Se propone como proyecto futuro estudiar la creación de una aplicación o sitio web de carácter más amplio, que reúna las todas las aplicaciones que se han creado por separado para abordar los diferentes temas que componen la asignatura Tratamiento Digital de Audio.
- Traducción de la web del tutorial y del texto de las applets al inglés, idioma cuyo conocimiento es esencial hoy en día. En los planes de estudio actuales la práctica de este idioma tiene una gran importancia y sería una ayuda para los alumnos; además, se impulsaría la difusión del tutorial fuera del ámbito local en el que ha sido creado,

ANEXO I: Distribución y despliegue del proyecto web

El proyecto web desarrollado se distribuye en un paquete compuesto por el siguiente conjunto de archivos organizados en la estructura de directorios indicada:



La página de bienvenida del tutorial corresponde al archivo "TutorialTecnicasDigitalizacionAudio.html". El directorio raíz donde se encuentran el resto de componentes del proyecto se denomina "TutorialTecnicasDigitalizacionAudio".

Las páginas que forman el tutorial corresponden al conjunto de archivos HTML nombrados como "paginaX.html". El archivo TDA.jar es el paquete ejecutable que contiene las applets incluidas en el proyecto. Los directorios contenidos clasifican diferentes tipos de archivos utilizados por el proyecto web y las applets. La carpeta "applets" contiene el código fuente (archivos "java") que implementa las aplicaciones (no son imprescindibles para su ejecución pero se ofrecen por si son de interés para el usuario), la carpeta "audio" contiene los ficheros de sonido en formato WAV que se pueden cargar como ejemplo en las aplicaciones, la carpeta "css" contiene las hojas de estilo del proyecto web, la

carpeta "js" contiene los archivos Javascript del proyecto web, que son dos: el archivo "script.js" que implementa la funcionalidad del sitio web y el archivo "jquery-1.3.2" que es la librería utilizada, la carpeta "icons" contiene los iconos y logotipos incluidos en el tutorial y finalmente, la carpeta "imagenes" contiene las figuras y gráficos que aparecen en el tutorial.

Ejecución de las applets

Las applets que incluye el proyecto web se distribuyen, como se ha explicado previamente, en el fichero jar "TDA". Este paquete contiene todos los archivos "class" necesarios para ejecutar cualquiera de las cuatro applets (Cuantificación, Sobremuestreo, Dither y Noise-Shaping), tanto los comunes como los específicos para cada una, además de la librería JFreeChart (contenida en el directorio "org") y los iconos de la interfaz en formato de imagen "GIF".

El paquete "TDA.jar" está firmado digitalmente para poder acceder a las funcionalidades que requieren permisos de seguridad especiales, como son, la carga de ficheros de audio desde el disco duro del usuario y guardar las gráficas de JFreeChart en formato PNG.

Los datos de la firma digital son los siguientes:

- Autor: Francisco Javier Martínez
- Departamento: Teoría de la Señal y Comunicaciones
- Organización: Universidad Carlos III de Madrid
- Ciudad: Leganés
- Provincia: Madrid
- País: ES (España)

La inclusión de esta firma implica que el usuario sea advertido de la misma en el momento previo a la ejecución del applet, y que tenga que aceptarla para continuar. En caso negativo, el applet no se iniciará.

Si se invocan las clases de arranque directamente, descomprimiendo el fichero TDA.jar, estas se ejecutarán sin la comprobación previa de la firma digital, pero las funcionalidades comentadas antes no estarán disponibles.

Para la ejecución correcta de las applets y el aprovechamiento máximo de sus posibilidades, es necesario configurar adecuadamente el parámetro "Xmx", que

es la cantidad de memoria RAM que la máquina virtual Java reserva al cargar el applet, cuyo valor mínimo debe ser de 500 Mb, el cual garantiza tener suficiente memoria para procesar cualquiera de las señales de audio de ejemplo incluidas. El valor por defecto de este parámetro es de 128 Mb, el cual es insuficiente, provocando errores de ejecución.

Las applets se invocan desde un archivo HTML con la siguiente etiqueta, en la que se especifican los siguientes datos: localización del fichero "TDA.jar", clase de arranque, altura y anchura del applet (900 x 510 px) y valor del parámetro "Xmx".

```
<applet archive="TDA.jar" code="QuantApplet.class" name="QuantApplet"
width="900" height="510">
  <param name="java_arguments" value="-Xmx500m" />
</applet>
```

```
<applet archive="TDA.jar" code="Oversampling.class"
name="Oversampling" width="900" height="510">
  <param name="java_arguments" value="-Xmx500m" />
</applet>
```

```
<applet archive="TDA.jar" code="DitherApplet.class"
name="DitherApplet" width="900" height="510">
  <param name="java_arguments" value="-Xmx500m" />
</applet>
```

```
<applet archive="TDA.jar" code="NoiseShapingApplet.class"
name="NoiseShapingApplet" width="900" height="510">
  <param name="java_arguments" value="-Xmx500m" />
</applet>
```


ANEXO II: Estructura de clases y métodos de las applets

Clases de arranque

QuantApplet.class	Oversampling.class	Dither.class	NoiseShapingApplet.class
Métodos init interfazGrafica extraccion transformacion transformacionInversa almacenamiento reproducir cuantificador crearVentanaGrafica ampliarVentanaGrafica actualizarInfoAudio interfazAbrir Clases internas OyenteMenuAbrir OyenteAbrir OyenteRep OyenteCuantificacion OyenteBarra OyenteDesplazamiento OyenteTabs	Métodos init interfazGrafica extraccion transformacion transformacionInversa almacenamiento reproducir cuantificador crearVentanaGrafica ampliarVentanaGrafica actualizarInfoAudio interfazAbrir resultadosNumericos Clases internas OyenteMenuAbrir OyenteAbrir OyenteRep OyenteCuantificacion OyenteSobremuestreo OyenteBarra OyenteDesplazamiento OyenteTabs	Métodos init interfazGrafica extraccion transformacion transformacionInversa almacenamiento reproducir cuantificador crearVentanaGrafica ampliarVentanaGrafica actualizarInfoAudio interfazAbrir resultadosNumericos Clases internas OyenteMenuAbrir OyenteAbrir OyenteRep OyenteCuantificacion OyenteBarra OyenteDesplazamiento OyenteTabs OyenteValoresOptimos OyenteDither	Métodos init interfazGrafica extraccion transformacion transformacionInversa almacenamiento reproducir cuantificador crearVentanaGrafica ampliarVentanaGrafica actualizarInfoAudio interfazAbrir resultadosNumericos verificarEstabilidad aplicarNoiseShaping Clases internas OyenteMenuAbrir OyenteAbrir OyenteRep OyenteCuantificacion OyenteBarra OyenteDesplazamiento OyenteTabs OyenteValoresOptimos OyenteNoiseShaping

Clases auxiliares

MetodosAuxiliares.class	Conversor.class	Sonido.class
Métodos cuantificarMuestra interpolar diezmar filtrar calcularPotenciaDEP enventanar obtenerFactorCuantificacion factorPonderacionVentanaHanning	Métodos transformacion transformacionInversa	Métodos repAudio
Graficas.class	GeneradorAleatorio.class	ArchivoAudio.class
Métodos panelGrafica panelGraficaMultiple panelGraficaPromedioDEP maximoSenal maximoDEP	Métodos arrayGaussFloat arrayUniformeFloat arrayTriangularFloat	Métodos ArchivoAudio

Librerías externas

FFT.class
Complex.class
JFreeChart

Bibliografía

Procesado digital de señal, audio y acústica

1. **Pohlman, Ken C.** *Principles of Digital Audio*. 2004. McGraw Hill.
2. **Zolzer, Udo.** *Digital Audio Signal Processing*. 1997. John Wiley & Sons.
3. **Watkinson, John.** *El arte del audio digital*. 1993. Paraninfo.
4. **Oppenheim, Alan V.** *Tratamiento de señales en tiempo discreto*. 1999. Pearson.
5. **Oppenheim, Alan V.** *Señales y sistemas*. 1998. Prentice Hall.
6. **Recuero, Manuel.** *Ingeniería Acústica*. 2000. Paraninfo.
7. **Smith III, Julius O.** *Spectral Audio Processing*. 2007. Stanford University.
<<http://ccrma.stanford.edu/~jos/sasp/>>
8. **Robinson, David J.M.** *The human auditory system*.
<http://www.mp3-tech.org/programmer/docs/human_auditory_system.pdf>

Técnicas de digitalización de audio

9. **De Koning, Dreten; Verhelst, Werner.** *On Psychoacoustic Noise Shaping For Audio Requantization*. IEEE International.
10. **Vest, Mikael.** *Psychoacoustic Dither in ADDA 2402*. 2003. Digital Audio Denmark.
11. **Dunn, Chris; Sandler, Mark.** *Psychoacoustically Optimal Sigma Delta Modulation*. 1997. AES.
<<http://www.scalatech.co.uk/papers/jaes497.pdf>>
12. **Norsworthy, Steven R.** *Optimal Nonrecursive Noise Shaping Filters For Oversampling Data Converters*. 2003. IEEE International.
13. **Aldrich, Nika.** *Dither Explained*. 2002. Cadenza Recording.
<<http://www.users.qwest.net/~volt42/cadenzarecording/DitherExplained.pdf>>
14. **Bohn, Dennis A.** *Digital Dharma of audio A/D Converters*. 1997. Rane.
<http://rane.com/notes137.html>
15. **Lyons, Richard; Yates, Randy.** *Reducing ADC Quantization Noise*. 2005. Microwaves and RF.
<<http://www.mwrf.com/Articles/ArticleID/10586/10586.html>>
16. **Wannamaker Robert A.; Lipshitz, Stanley P.; Vanderkooy, John; Wright J. Nelson.** *A Theory of Nonsubtractive Dither*. 2000. IEEE International.
<<http://audiolab.uwaterloo.ca/~rob/abstracts/ieee.pdf>>
17. **Mínguez, Antonio; Recuero Manuel.** *Moldeado espectral de ruido acústico*. Acústica 2000.
<<http://www.sea-acustica.es/publicaciones/4350cb005.pdf>>

18. **Tomarakos, John.** *The relationship of Dinamic Range to Data Word Size in Digital Audio Processing.* Analog Devices.
<http://www.analog.com/en/embedded-processing-dsp/sharc/content/relationship_data_word_size_dynamic_range/fca.html>
19. **Wooley, Bruce A.** *Cascaded Noise-Shaping for Oversampling A/D and D/A conversion.* 2005. Stanford Universtiy.
<<http://www.ewh.ieee.org/r6/scv/ssc/Nov1804.pdf>>
20. **Valle, Mauricio.** *Teoría de Cuantización, Conversión Analógica-Digital y Digital-Analógica.*
21. **Kester, Walt.** *Taking the mistery out of the infamous formula "SNR= 6.02 +1.76 dB" and why you should care.* 2005. Analog Devices.
<<http://www.analog.com/static/imported-files/tutorials/MT-001.pdf>>
22. **Kiss, Peter; Arias, Jesus; Li, Dandan; Boccuzzi, Vito.** *Stable High-Order Delta-Sigma Digital-to-Analog Converters.* 2004. IEEE International.
23. **Borque Palacín, Alfredo.** *Introducción a la transmisión digital de señales de audio y video.* 2007. Cypsela.
<http://www.cypsela.es/especiales/pdf217/transmision_digital.pdf>
24. **Kirac, A; Vaidyanathan P.P.** *Dithering in Lattice Quantization.* 1995. IEEE.
25. **Estévez Báez, Mario; Machado García, Mario; Estévez Carrera, José M.** *El periodograma como método para la estimación de la densidad espectral de potencia.* 2008.

Java y desarrollo web

26. **Sánchez, J. ; Huecas, G.** *Java 2: Iniciación y Referencia.* 2001. McGraw Hill.
27. **Pfisterer, Matthias; Bomers, Florian.** *Applet Programing with Java Sound API.* 2001. JavaOne.
<<http://www.jsresources.org/apps/1378.pdf>>
28. **Baldwin, Richard G.** *Java Sound, An Introduction.* 2003.
<<http://www.developer.com/java/other/article.php/1565671>>
29. **Prades del Valle, Carlos.** *Sonido en Java.* 2001.
<<http://cprades.eresmas.com/Tecnica/sonidoenjava.pdf>>
30. **Canales Mora, Roberto.** *Gráficas en Java con JFreeChart.*
31. **Gilbert , David.** *JFreeChart Developer Guide.*
32. **Eguíluz Pérez, Javier.** *Introducción a XHTML.* Librosweb.
<<http://www.librosweb.es/xhtml/index.html>>
33. **Eguíluz Pérez, Javier.** *Introducción a CSS.* Librosweb.
< <http://www.librosweb.es/css/index.html>>
34. **Eguíluz Pérez, Javier.** *Introducción a Javascript.* Librosweb.
<<http://www.librosweb.es/javascript/index.html>>
35. **Moreno, Luciano.** *Curso práctico de diseño web.* DesarrolloWeb.
<<http://www.desarrolloweb.com/manuales/47/>>

36. **Manchón, Eduardo.** *Principios generales de usabilidad en sitios web.*

DesarrolloWeb.

<<http://www.desarrolloweb.com/articulos/1133.php>>

Tele-educación y e-learning

37. **García Peñalvo, Francisco José.** *Estado actual de los sistemas e-learning.*

2005. Universidad de Salamanca.

<http://www.usal.es/~teoriaeducacion/rev_numero_06_2/n6_02_art_garcia_penalvo.htm>

38. **Rosenberg, Marc J.** *What lies beyond E-learning?* 2006. Learning Circuits.

<http://www.marcrosenberg.com/images/What_Lies_Beyond_E-Learning_ASTD.pdf>

39. **Sánchez Torrubia, M.G.; Torres Blanc, C.; Giménez Martínez, V.**

Tutoriales interactivos en Java: una excelente herramienta para la enseñanza y el aprendizaje de los algoritmos de grafos. 2007. Universidad Politécnica de Madrid.

<http://quinto.fmetsia.upm.es/archivo/mesa_incorp_nuevas_tecnologias_formac_presencial.pdf>

40. **Guillermard, Mijail; Ruwwe, Christian; Zolzer, Udo.** *Audio effect applets for music processing.* 2006. Helmut-Schmidt University.

<<http://www-mmdb.iai.uni-bonn.de/konferenzen/dag-sitzung06/audioApplets.pdf>>

41. **Megia Miranda, Marisa.** *Tele-educación: el futuro de la educación.*

Índice de figuras

Figura 1-1 Audio Effect Applets for Music Processing - Cuantificación.....	24
Figura 1-2 Audio Effect Applets for Music Processing - Dither.....	24
Figura 1-3 Audio Effect Applets for Music Processing – Noise Shaping.....	25
Figura 1-4 Audio Effect Applets for Music Processing - Oversampling x 8.....	25
Figura 1-5 Audio Effect Applets for Music Processing - Oversampling x 4.....	26
Figura 1-6 Virtual Laboratories.....	29
Figura 2-1 Coste computacional FFT.....	45
Figura 2-2 Ventana Hann - Tiempo.....	50
Figura 2-3 Ventana Hann - Frecuencia.....	50
Figura 2-4 Tipos de filtros.....	53
Figura 2-5 Estructura filtro FIR.....	55
Figura 2-6 Estructura filtro IIR.....	56
Figura 2-7 Filtro IIR 1/2 (Magnitud).....	59
Figura 2-8 Filtro IIR 1/2 (Fase).....	59
Figura 2-9 Filtro IIR 1/2 (Diagrama de polos y ceros).....	59
Figura 2-10 Filtro IIR 1/4 (Magnitud).....	60
Figura 2-11 Filtro IIR 1/4 (Fase).....	60
Figura 2-12 Filtro IIR 1/6 (Diagrama de polos y ceros).....	60
Figura 2-13 Filtro IIR 1/8 (Magnitud).....	61
Figura 2-14 Filtro IIR 1/8 (Fase).....	61
Figura 2-15 Filtro IIR 1/8 (Diagrama de polos y ceros).....	61
Figura 2-16 Ejemplo señal analógica.....	62
Figura 2-17 Ejemplo señal digital.....	62
Figura 2-19 Paso de señal analógica a muestreada.....	63
Figura 2-18 Diagrama Conversión A / D.....	63
Figura 2-20 Paso de dominio analógico a digital (espectro).....	64
Figura 2-21 Aliasing.....	65
Figura 2-22 Regeneración de señal digital afectada por ruido.....	69
Figura 2-23 Esquema de cuantificación.....	74
Figura 2-24 Tipos de cuantificación uniforme.....	76
Figura 2-25 Cuantificador logarítmico.....	77
Figura 2-26 Tono cuantificado frente a un tono puro.....	81
Figura 2-28 Efecto del sobremuestreo.....	84
Figura 2-27 Diagrama de la interpolación.....	84
Figura 2-29 Filtro antialiasing de pendiente elevada.....	86
Figura 2-30 Filtro antialiasing de pendiente suave.....	86
Figura 2-31 Tono de 800 Hz en tiempo y frecuencia.....	89
Figura 2-32 Tono de 800 Hz en tiempo y frecuencia.....	89
Figura 2-33 Tono de 800 Hz con dither en tiempo y frecuencia.....	89
Figura 2-34 Diagrama dither substractivo.....	91

Figura 2-35 Dither diagrama dither no substractivo	92
Figura 2-36 Función G_v' para dither nulo	94
Figura 2-37 Función densidad de probabilidad uniforme	95
Figura 2-38 Primera derivada de $G_v(u)$ de fdp uniforme.....	95
Figura 2-39 Segunda derivada de $G_v(u)$ de fdp uniforme.....	95
Figura 2-40 Primera derivada de $G_v(u)$ de fdp triangular	96
Figura 2-41 Segunda derivada de $G_v(u)$ de fdp triangular	96
Figura 2-42 Tercera derivada de $G_v(u)$ de fdp triangular	97
Figura 2-43 Diagrama de Noise Shaping de orden N	100
Figura 2-44 Diagrama de Noise Shaping de orden 1	102
Figura 2-45 Diagrama de Noise Shaping de orden 2	102
Figura 2-46 Diagrama de Noise - Shaping de orden 3	103
Figura 2-47 Respuesta en frecuencia (Magnitud) de un filtro de Noise - Shaping de orden 1.....	103
Figura 2-48 Respuesta en frecuencia (Fase) de un filtro de Noise - Shaping de orden 1.....	104
Figura 2-49 Diagrama de polos y ceros de un filtro de Noise-Shaping de orden 1	104
Figura 2-50 Respuesta en frecuencia (Magnitud) de un filtro de Noise - Shaping de orden 2.....	104
Figura 2-51 Respuesta en frecuencia (Fase) de un filtro de Noise - Shaping de orden 2.....	105
Figura 2-52 Diagrama de polos y ceros de un filtro de Noise-Shaping de orden 2	105
Figura 2-53 Respuesta en frecuencia (Magnitud) de un filtro de Noise - Shaping de orden 3.....	105
Figura 2-54 Respuesta en frecuencia (Fase) de un filtro de Noise - Shaping de orden 2.....	106
Figura 2-55 Diagrama de polos y ceros de un filtro de Noise-Shaping de orden 3	106
Figura 2-56 Comparativa de funciones de Noise - Shaping.....	106
Figura 2-57 Señal de voz cuantificada con 16 bits	108
Figura 2-58 Señal de voz cuantificada con 4 bits aplicando Noise - Shaping de orden 1.....	108
Figura 2-59 Señal de voz cuantificada con 4 bits aplicando Noise - Shaping de orden 2.....	108
Figura 2-60 Señal de voz cuantificada con 4 bits aplicando Noise - Shaping de orden 3.....	109
Figura 2-61 Señal de voz cuantificada con 4 bits aplicando Noise - Shaping de orden 4.....	109
Figura 2-62 Curvas de audición (Equal - Loudness curves)	112
Figura 2-63 E-weighting curve	115
Figura 3-1 Test Acid para Firefox 3	130

Figura 3-2 Test Acid para Internet Explorer 7	130
Figura 3-3 Gráfico Google Analytics del número de visitas recibidas en eje temporal	140
Figura 3-4 Gráfico Google Analytics de visitas en función del navegador web	140
Figura 3-5 Gráfico Google Analytics de la procedencia geográfica de las visitas ...	141
Figura 3-6 Ejemplos de gráficas JFreeChart	150
Figura 4-1 Menú de Inicio	163
Figura 4-2 Vista principal	163
Figura 4-3 Estudio práctico (aplicación visible)	164
Figura 4-4 Estudio práctico (instrucciones visibles)	164
Figura 4-5 Esquema de organización del sitio web (I)	165
Figura 4-6 Esquema de organización del sitio web (II)	166
Figura 4-7 Esquema de organización del sitio web (III)	167
Figura 4-8 Esquema de organización del sitio web (IV)	168
Figura 4-9 Estructura básica de las aplicaciones	178
Figura 4-10 Secuencia básica de manejo	179
Figura 4-11 Cabecera	179
Figura 4-12 Cuadro de diálogo "Abrir fichero de audio"	180
Figura 4-13 Secuencia de apertura de un fichero de audio	181
Figura 4-14 Resultados gráficos	183
Figura 4-15 Resultados auditivos	184
Figura 4-16 Resultados numéricos	185
Figura 4-17 Configuración de parámetros (Paso 1)	187
Figura 4-18 Configuración de parámetros (Paso 2)	187
Figura 4-19 Vista principal del applet de cuantificación	188
Figura 4-20 Vista principal del applet de sobremuestreo	188
Figura 4-21 Vista principal del applet de dither	189
Figura 4-22 Vista principal del applet de Noise - Shaping	189
Figura 4-23 Diagrama de sobremuestreo	191
Figura 4-24 Gráfica de sobremuestreo (eA / eB1)	192
Figura 4-26 Gráfica de sobremuestreo (eA / eB2)	192
Figura 4-25 Gráfica de DEP Promedio de Sobremuestreo	192
Figura 4-27 Gráfica de sobremuestreo (eA / eB3)	193
Figura 4-28 Aplicación no óptima de dither (en el espectro)	195
Figura 4-29 Aplicación óptima de dither (en el espectro)	195
Figura 4-30 Efecto del dither en el tiempo	196
Figura 4-31 Gráfica de Noise - Shaping (eA / eB1) espectro	197
Figura 4-32 Gráfica de Noise - Shaping DEP Promedio (eA / eB1) en espectro	198
Figura 4-33 Gráfica de Noise - Shaping (eA / eB2) en espectro	198
Figura 4-34 Gráfica de Noise - Shaping (eA / eB3) en tiempo	198
Figura 4-35 Gráfica de Noise - Shaping (eA / eB3) en espectro	199
Figura 4-36 Diagrama de organización de clases	203

Pliego de condiciones

Este documento contiene las condiciones legales que guiarán la realización, en este proyecto, de un Tutorial Web de Técnicas de Digitalización de Audio. En lo que sigue, se supondrá que el proyecto ha sido encargado por una empresa cliente a una empresa consultora con la finalidad de realizar dicho sistema. Dicha empresa ha debido desarrollar una línea de investigación con objeto de elaborar el proyecto. Esta línea de investigación, junto con el posterior desarrollo de los programas está amparada por las condiciones particulares del siguiente pliego.

Supuesto que la utilización industrial de los métodos recogidos en el presente proyecto ha sido decidida por parte de la empresa cliente o de otras, la obra a realizar se regulará por las siguientes:

Condiciones generales

1. La modalidad de contratación será el concurso. La adjudicación se hará, por tanto, a la proposición más favorable sin atender exclusivamente al valor económico, dependiendo de las mayores garantías ofrecidas. La empresa que somete el proyecto a concurso se reserva el derecho a declararlo desierto.
2. El montaje y mecanización completa de los equipos que intervengan será realizado totalmente por la empresa licitadora.
3. En la oferta, se hará constar el precio total por el que se compromete a realizar la obra y el tanto por ciento de baja que supone este precio en relación con un importe límite si este se hubiera fijado.
4. La obra se realizará bajo la dirección técnica de un Ingeniero Superior de Telecomunicación, auxiliado por el número de Ingenieros Técnicos y Programadores que se estime preciso para el desarrollo de la misma.
5. Aparte del Ingeniero Director, el contratista tendrá derecho a contratar al resto del personal, pudiendo ceder esta prerrogativa a favor del Ingeniero Director, quien no estará obligado a aceptarla.
6. El contratista tiene derecho a sacar copias a su costa de los planos, pliego de condiciones y presupuestos. El Ingeniero autor del proyecto autorizará con su firma las copias solicitadas por el contratista después de confrontarlas.
7. Se abonará al contratista la obra que realmente ejecute con sujeción al proyecto que sirvió de base para la contratación, a las modificaciones autorizadas por la superioridad o a las órdenes que con arreglo a sus facultades le hayan comunicado por escrito al Ingeniero Director de obras siempre que dicha obra se haya ajustado a los preceptos de los pliegos de condiciones, con arreglo a los cuales, se harán las modificaciones y la valoración de las diversas unidades sin que el importe total pueda exceder de los presupuestos aprobados. Por consiguiente, el número de unidades que se consignan en el proyecto o en el presupuesto, no podrá servirle de fundamento para entablar reclamaciones de ninguna clase, salvo en los casos de rescisión.

8. Tanto en las certificaciones de obras como en la liquidación final, se abonarán los trabajos realizados por el contratista a los precios de ejecución material que figuran en el presupuesto para cada unidad de la obra.
9. Si excepcionalmente se hubiera ejecutado algún trabajo que no se ajustase a las condiciones de la contrata pero que sin embargo es admisible a juicio del Ingeniero Director de obras, se dará conocimiento a la Dirección, proponiendo a la vez la rebaja de precios que el Ingeniero estime justa y si la Dirección resolviera aceptar la obra, quedará el contratista obligado a conformarse con la rebaja acordada.
10. Cuando se juzgue necesario emplear materiales o ejecutar obras que no figuren en el presupuesto de la contrata, se evaluará su importe a los precios asignados a otras obras o materiales análogos si los hubiere y cuando no, se discutirán entre el Ingeniero Director y el contratista, sometiéndolos a la aprobación de la Dirección. Los nuevos precios convenidos por uno u otro procedimiento, se sujetarán siempre al establecido en el punto anterior.
11. Cuando el contratista, con autorización del Ingeniero Director de obras, emplee materiales de calidad más elevada o de mayores dimensiones de lo estipulado en el proyecto, o sustituya una clase de fabricación por otra que tenga asignado mayor precio o ejecute con mayores dimensiones cualquier otra parte de las obras, o en general, introduzca en ellas cualquier modificación que sea beneficiosa a juicio del Ingeniero Director de obras, no tendrá derecho sin embargo, sino a lo que le correspondería si hubiera realizado la obra con estricta sujeción a lo proyectado y contratado.
12. Las cantidades calculadas para obras accesorias, aunque figuren por partidaalzada en el presupuesto final (general), no serán abonadas sino a los precios de la contrata, según las condiciones de la misma y los proyectos particulares que para ellas se formen, o en su defecto, por lo que resulte de su medición final.
13. El contratista queda obligado a abonar al Ingeniero autor del proyecto y director de obras así como a los Ingenieros Técnicos, el importe de sus respectivos honorarios facultativos por formación del proyecto, dirección técnica y administración en su caso, con arreglo a las tarifas y honorarios vigentes.
14. Concluida la ejecución de la obra, será reconocida por el Ingeniero Director que a tal efecto designe la empresa.
15. La garantía definitiva será del 4% del presupuesto y la provisional del 2%.
16. La forma de pago será por certificaciones mensuales de la obra ejecutada, de acuerdo con los precios del presupuesto, deducida la baja si la hubiera.
17. La fecha de comienzo de las obras será a partir de los 15 días naturales del replanteo oficial de las mismas y la definitiva, al año de haber ejecutado la provisional, procediéndose si no existe reclamación alguna, a la reclamación de la fianza.

18. Si el contratista al efectuar el replanteo, observase algún error en el proyecto, deberá comunicarlo en el plazo de quince días al Ingeniero Director de obras, pues transcurrido ese plazo será responsable de la exactitud del proyecto.
19. El contratista está obligado a designar una persona responsable que se entenderá con el Ingeniero Director de obras, o con el delegado que éste designe, para todo relacionado con ella. Al ser el Ingeniero Director de obras el que interpreta el proyecto, el contratista deberá consultarle cualquier duda que surja en su realización.
20. Durante la realización de la obra, se girarán visitas de inspección por personal facultativo de la empresa cliente, para hacer las comprobaciones que se crean oportunas. Es obligación del contratista, la conservación de la obra ya ejecutada hasta la recepción de la misma, por lo que el deterioro parcial o total de ella, aunque sea por agentes atmosféricos u otras causas, deberá ser reparado o reconstruido por su cuenta.
21. El contratista, deberá realizar la obra en el plazo mencionado a partir de la fecha del contrato, incurriendo en multa, por retraso de la ejecución siempre que éste no sea debido a causas de fuerza mayor. A la terminación de la obra, se hará una recepción provisional previo reconocimiento y examen por la dirección técnica, el depositario de efectos, el interventor y el jefe de servicio o un representante, estampando su conformidad el contratista.
22. Hecha la recepción provisional, se certificará al contratista el resto de la obra, reservándose la administración el importe de los gastos de conservación de la misma hasta su recepción definitiva y la fianza durante el tiempo señalado como plazo de garantía. La recepción definitiva se hará en las mismas condiciones que la provisional, extendiéndose el acta correspondiente. El Director Técnico propondrá a la Junta Económica la devolución de la fianza al contratista de acuerdo con las condiciones económicas legales establecidas.
23. Las tarifas para la determinación de honorarios, reguladas por orden de la Presidencia del Gobierno el 19 de Octubre de 1961, se aplicarán sobre el denominado en la actualidad "Presupuesto de Ejecución de Contrata" y anteriormente llamado "Presupuesto de Ejecución Material" que hoy designa otro concepto.

Condiciones particulares

1. La empresa consultora, que ha desarrollado el presente proyecto, lo entregará a la empresa cliente bajo las condiciones generales ya formuladas, debiendo añadirse las siguientes condiciones particulares:
2. La propiedad intelectual de los procesos descritos y analizados en el presente trabajo, pertenece por entero a la empresa consultora representada por el Ingeniero Director del Proyecto. La empresa consultora se reserva el derecho a la utilización total o parcial de los resultados de la investigación realizada para desarrollar el siguiente proyecto, bien para su publicación o bien para su uso en trabajos o proyectos posteriores, para la misma empresa cliente o para otra.

3. Cualquier tipo de reproducción aparte de las reseñadas en las condiciones generales, bien sea para uso particular de la empresa cliente, o para cualquier otra aplicación, contará con autorización expresa y por escrito del Ingeniero Director del Proyecto, que actuará en representación de la empresa consultora. En la autorización se ha de hacer constar la aplicación a que se destinan sus reproducciones así como su cantidad.
4. En todas las reproducciones se indicará su procedencia, explicitando el nombre del proyecto, nombre del Ingeniero Director y de la empresa consultora.
5. Si el proyecto pasa la etapa de desarrollo, cualquier modificación que se realice sobre él, deberá ser notificada al Ingeniero Director del Proyecto y a criterio de éste, la empresa consultora decidirá aceptar o no la modificación propuesta.
6. Si la modificación se acepta, la empresa consultora se hará responsable al mismo nivel que el proyecto inicial del que resulta el añadirla.
7. Si la modificación no es aceptada, por el contrario, la empresa consultora declinará toda responsabilidad que se derive de la aplicación o influencia de la misma.
8. Si la empresa cliente decide desarrollar industrialmente uno o varios productos en los que resulte parcial o totalmente aplicable el estudio de este proyecto, deberá comunicarlo a la empresa consultora.
9. La empresa consultora no se responsabiliza de los efectos laterales que se puedan producir en el momento en que se utilice la herramienta objeto del presente proyecto para la realización de otras aplicaciones.
10. La empresa consultora tendrá prioridad respecto a otras en la elaboración de los proyectos auxiliares que fuese necesario desarrollar para dicha aplicación industrial, siempre que no haga explícita renuncia a este hecho. En este caso, deberá autorizar expresamente los proyectos presentados por otros.
11. El Ingeniero Director del presente proyecto, será el responsable de la dirección de la aplicación industrial siempre que la empresa consultora lo estime oportuno. En caso contrario, la persona designada deberá contar con la autorización del mismo, quien delegará en él las responsabilidades que ostente.

Presupuesto

Ejecución material

Ordenador personal portátil	900 €
Material de oficina.....	30 €
Alquiler servidor web de desarrollo	150 €
Conexión a Internet ADSL durante 12 meses	360 €

Gastos generales

16 % sobre ejecución material	230 €
-------------------------------------	-------

Beneficio industrial

6 % sobre ejecución material	86 €
------------------------------------	------

Honorarios (sueldo Ingeniero Técnico)

12 meses x 20 días/ mes x 8 horas/día x 15 €/hora	28800 €
---	---------

Material fungible

Gastos de impresión	50 €
Gastos de encuadernación	200 €

Subtotal del presupuesto

Subtotal del presupuesto	30806 €
--------------------------------	---------

I.V.A aplicable

16 % sobre Subtotal del presupuesto.....	4928 €
--	--------

Total presupuesto

Total presupuesto	35734 €
-------------------------	---------

Leganés, Mayo de 2009

El Ingeniero Jefe del Proyecto

Fdo: Francisco Javier Martínez Mencías

Ingeniero Técnico de Telecomunicación

